

GAME SYSTEM, PROGRAM AND INFORMATION STORAGE MEDIUM

Patent number: JP2002032780
Publication date: 2002-01-31
Inventor: KIKKO SHIGERU
Applicant: NAMCO LTD
Classification:
- international: G06T15/00; A63F13/00; G06T15/70;
G06T17/40
- european:
Application number: JP20010134264 20010501
Priority number(s): JP20000136835 20000510;
JP20010134264 20010501

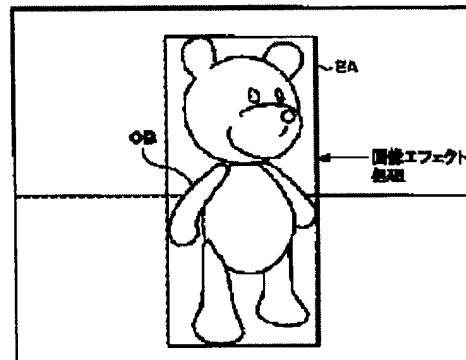
Report a data error here

Abstract of JP2002032780

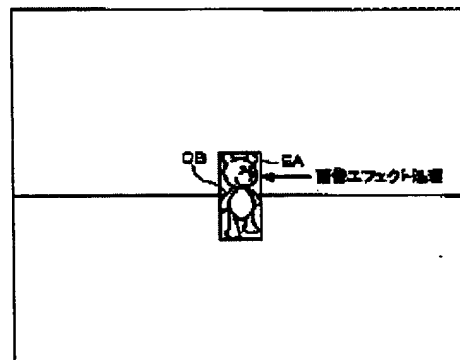
PROBLEM TO BE SOLVED: To provide a game system capable of efficiently realizing image effect processing, a program and an information storage medium.

SOLUTION: An effect area EA which includes the image of an object OB that has been subjected to perspective transformation and whose size changes in accordance with the size of the OB after the perspective transformation is set, and the image of the EA is subjected to image effect processing. The EA is set on the basis of the top of the OB after the perspective transformation and the coordinates of the top of a simple object. The image of the effect area EA is mapped on a virtual polygon in the form of the EA with a bilinear filtering method while shifting texture coordinates. The information of the EA image is set as the index number of a lookup table LUT for index color texture mapping and index color texture mapping is applied to the virtual polygon by using the LUT. Image effect processing for enhancing a counter is applied to the object.

(A) 視点からの距離が近い場合



(B) 視点からの距離が遠い場合



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-32780

(P2002-32780A)

(43) 公開日 平成14年1月31日 (2002.1.31)

(51) Int.Cl. ⁷	識別記号	F I	テマコード* (参考)
G 0 6 T 15/00	3 0 0	G 0 6 T 15/00	3 0 0 2 C 0 0 1
	1 0 0		1 0 0 A 5 B 0 5 0
A 6 3 F 13/00		A 6 3 F 13/00	C 5 B 0 8 0
G 0 6 T 15/70		G 0 6 T 15/70	A
17/40		17/40	A
審査請求 有 請求項の数19 O L (全 24 頁)			

(21) 出願番号 特願2001-134264 (P2001-134264)

(22) 出願日 平成13年5月1日 (2001.5.1)

(31) 優先権主張番号 特願2000-136835 (P2000-136835)

(32) 優先日 平成12年5月10日 (2000.5.10)

(33) 優先権主張国 日本 (J P)

(71) 出願人 000134855

株式会社ナムコ

東京都大田区多摩川2丁目8番5号

(72) 発明者 橋高 繁

東京都大田区多摩川2丁目8番5号 株式

会社ナムコ内

(74) 代理人 100090387

弁理士 布施 行夫 (外2名)

最終頁に続く

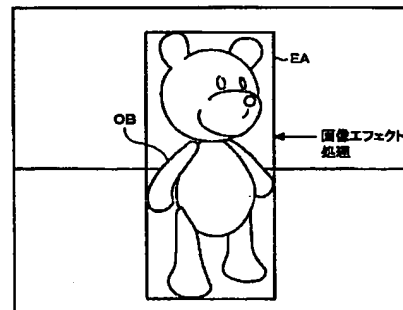
(54) 【発明の名称】 ゲームシステム、プログラム及び情報記憶媒体

(57) 【要約】

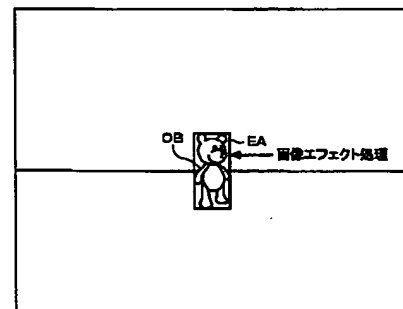
【課題】 効率の良い画像エフェクト処理を実現できるゲームシステム、プログラム及び情報記憶媒体を提供すること。

【解決手段】 透視変換後のオブジェクトOBの画像を内包し透視変換後のOBの大きさに応じてその大きさが変化するエフェクト領域EAを設定し、EAの画像に対して画像エフェクト処理を施す。透視変換後のOBの頂点や簡易オブジェクトの頂点の座標に基づきEAを設定する。エフェクト領域EAの画像を、テクスチャ座標をシフトさせながらバイリニアフィルタ方式でEAの形状の仮想ポリゴンにマッピングする。EAの画像の情報を、インデックスカラー・テクスチャマッピング用のルックアップテーブルLUTのインデックス番号として設定し、そのLUTを用いて、仮想ポリゴンにインデックスカラー・テクスチャマッピングを行う。オブジェクトに輪郭線を強調する画像エフェクト処理を行う。

(A) 視点からの距離が近い場合



(B) 視点からの距離が遠い場合



【特許請求の範囲】

【請求項1】 画像生成を行うゲームシステムであって、
透視変換後のオブジェクトの画像の全部又は一部を内包し透視変換後のオブジェクトの大きさに応じてその大きさが変化するエフェクト領域を設定する手段と、
前記エフェクト領域の画像に対して画像エフェクト処理を施す手段と、
オブジェクト空間内の所与の視点での画像を生成する手段と、
を含むことを特徴とするゲームシステム。

【請求項2】 請求項1において、
透視変換後のオブジェクトの定義点の座標に基づいて前記エフェクト領域が設定されることを特徴とするゲームシステム。

【請求項3】 請求項1において、
オブジェクトに対して簡易オブジェクトが設定される場合において、透視変換後の前記簡易オブジェクトの定義点の座標に基づいて前記エフェクト領域が設定されることを特徴とするゲームシステム。

【請求項4】 請求項1乃至3のいずれかにおいて、
前記画像エフェクト処理が、
前記エフェクト領域の画像を用いた前記エフェクト領域の形状の仮想オブジェクトへのテクスチャマッピング処理であることを特徴とするゲームシステム。

【請求項5】 請求項4において、
前記エフェクト領域の画像を、テクスチャ座標をシフトさせながらテクセル補間方式で、前記エフェクト領域の形状の仮想オブジェクトにテクスチャマッピングすることを特徴とするゲームシステム。

【請求項6】 請求項5において、
前記テクセル補間方式のテクスチャマッピングがバイリニアフィルタ方式のテクスチャマッピングであることを特徴とするゲームシステム。

【請求項7】 請求項4において、
前記エフェクト領域の画像の情報を、インデックスカラー・テクスチャマッピング用のルックアップテーブルのインデックス番号として設定し、該ルックアップテーブルを用いて、前記エフェクト領域の形状の仮想オブジェクトにインデックスカラー・テクスチャマッピングを行うことを特徴とするゲームシステム。

【請求項8】 請求項4において、
前記エフェクト領域の画像に加工処理を施すことで得られた画像を、前記エフェクト領域の形状の仮想オブジェクトにテクスチャマッピングすることを特徴とするゲームシステム。

【請求項9】 請求項1乃至8のいずれかにおいて、
前記画像エフェクト処理が、
前記オブジェクトの画像に輪郭線の画像を付加するための処理であることを特徴とするゲームシステム。

【請求項10】 コンピュータにより使用可能なプログラムあって、

透視変換後のオブジェクトの画像の全部又は一部を内包し透視変換後のオブジェクトの大きさに応じてその大きさが変化するエフェクト領域を設定する手段と、
前記エフェクト領域の画像に対して画像エフェクト処理を施す手段と、
オブジェクト空間内の所与の視点での画像を生成する手段と、

をコンピュータに実行させるための処理ルーチンを含むことを特徴とするプログラム。

【請求項11】 請求項10において、
透視変換後のオブジェクトの定義点の座標に基づいて前記エフェクト領域が設定されることを特徴とするプログラム。

【請求項12】 請求項10において、
オブジェクトに対して簡易オブジェクトが設定される場合において、透視変換後の前記簡易オブジェクトの定義点の座標に基づいて前記エフェクト領域が設定されることを特徴とするプログラム。

【請求項13】 請求項10乃至12のいずれかにおいて、
前記画像エフェクト処理が、
前記エフェクト領域の画像を用いた前記エフェクト領域の形状の仮想オブジェクトへのテクスチャマッピング処理であることを特徴とするプログラム。

【請求項14】 請求項13において、
前記エフェクト領域の画像を、テクスチャ座標をシフトさせながらテクセル補間方式で、前記エフェクト領域の形状の仮想オブジェクトにテクスチャマッピングすることを特徴とするプログラム。

【請求項15】 請求項14において、
前記テクセル補間方式のテクスチャマッピングがバイリニアフィルタ方式のテクスチャマッピングであることを特徴とするプログラム。

【請求項16】 請求項13において、
前記エフェクト領域の画像の情報を、インデックスカラー・テクスチャマッピング用のルックアップテーブルのインデックス番号として設定し、該ルックアップテーブルを用いて、前記エフェクト領域の形状の仮想オブジェクトにインデックスカラー・テクスチャマッピングを行うことを特徴とするプログラム。

【請求項17】 請求項13において、
前記エフェクト領域の画像に加工処理を施すことで得られた画像を、前記エフェクト領域の形状の仮想オブジェクトにテクスチャマッピングすることを特徴とするプログラム。

【請求項18】 請求項10乃至17のいずれかにおいて、
前記画像エフェクト処理が、

前記オブジェクトの画像に輪郭線の画像を付加するための処理であることを特徴とするプログラム。

【請求項19】 コンピュータにより使用可能な情報記憶媒体であって、請求項10乃至18のいずれかのプログラムを含むことを特徴とする情報記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ゲームシステム、プログラム及び情報記憶媒体に関する。

【0002】

【背景技術及び発明が解決しようとする課題】従来より、仮想的な3次元空間であるオブジェクト空間内の所与の視点から見える画像を生成するゲームシステムが知られており、いわゆる仮想現実を体験できるものとして人気が高い。レーシングゲームを楽しむことができるゲームシステムを例にとれば、プレーヤは、車（オブジェクト）を操作してオブジェクト空間内で走行させ、他のプレーヤやコンピュータが操作する車と競争することで3次元ゲームを楽しむ。

【0003】さて、このようなゲームシステムでは、プレーヤの仮想現実感の向上のために、よりリアルで高品質な画像を生成することが重要な技術的課題になっている。このため、生成される画像に対して、輪郭線を付加する処理（輪郭線を強調する処理）、ぼかし処理などの種々の画像エフェクト処理を施すことが望まれる。

【0004】しかしながら、このような画像エフェクト処理の負荷は一般的に非常に重い。また、この種のゲームシステムでは、1フレーム（1/60秒）内で1画面分の描画処理を全て完了しなければならないというリアルタイム処理の要請もある。従って、画像エフェクト処理の負荷の軽減化や効率化を如何にして実現するかが重要な課題となる。

【0005】本発明は、以上のような課題に鑑みてなされたものであり、その目的とするところは、効率の良い画像エフェクト処理を実現できるゲームシステム、プログラム及び情報記憶媒体を提供することにある。

【0006】

【課題を解決するための手段】上記課題を解決するために、本発明は、画像生成を行うゲームシステムであって、透視変換後のオブジェクトの画像の全部又は一部を内包し透視変換後のオブジェクトの大きさに応じてその大きさが変化するエフェクト領域を設定する手段と、前記エフェクト領域の画像に対して画像エフェクト処理を施す手段と、オブジェクト空間内の所与の視点での画像を生成する手段とを含むことを特徴とする。また本発明に係る情報記憶媒体は、コンピュータにより使用可能な情報記憶媒体であって、上記手段を実行するためのプログラムを含むことを特徴とする。また本発明に係るプログラムは、コンピュータにより使用可能なプログラム（搬送波に具現化されるプログラムを含む）であって、

上記手段を実行するための処理ルーチンを含むことを特徴とする。

【0007】本発明によれば、透視変換後のオブジェクトの画像（或いはその一部）を内包するエフェクト領域が設定され、このエフェクト領域の画像に対して画像エフェクト処理を施される。そして、このエフェクト領域は、透視変換後のオブジェクトの大きさに応じてその大きさが変化する。従って、本発明によれば、画面内でのオブジェクトの占有面積が変化した場合に、それに応じてエフェクト領域の大きさも変化する。従って、透視変換後のオブジェクトの大きさに応じた最適な範囲で画像エフェクト処理が施されるようになり、画像エフェクト処理の負荷を軽減できる。

【0008】また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、透視変換後のオブジェクトの定義点の座標に基づいて前記エフェクト領域が設定されることを特徴とする。このようにすれば、透視変換後のオブジェクトの定義点（制御点、頂点等）の座標の例えば最大値、最小値などを求めるだけで、透視変換後のオブジェクトの大きさに応じてその大きさが変化するエフェクト領域を設定できるようになる。そして、エフェクト領域の大きさを、透視変換後のオブジェクトの大きさに応じて最適に小さくできるため、画像エフェクト処理の負荷を軽減できる。

【0009】また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、オブジェクトに対して簡易オブジェクトが設定される場合において、透視変換後の前記簡易オブジェクトの定義点の座標に基づいて前記エフェクト領域が設定されることを特徴とする。このようにすれば、透視変換後の簡易オブジェクトの定義点（制御点、頂点等）の座標の例えば最大値、最小値などを求めるだけで、透視変換後のオブジェクトの大きさに応じてその大きさが変化するエフェクト領域を設定できるようになる。そして、最大値、最小値などを求める処理の対象は、頂点数の少ない簡易オブジェクトの頂点の座標であるため、処理負荷が少なく済むという利点がある。

【0010】また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、前記画像エフェクト処理が、前記エフェクト領域の画像を用いた前記エフェクト領域の形状の仮想オブジェクトへのテクスチャマッピング処理であることを特徴とする。このようにすれば、エフェクト領域の画像を用いた仮想オブジェクトへのテクスチャマッピング処理を行うだけで、エフェクト領域の画像に画像エフェクト処理を施すことができるため、画像エフェクト処理の負荷を軽減できる。

【0011】なお仮想オブジェクトは、ポリゴンなどのプリミティブ面であることが望ましいが、立体的なオブジェクトであってもよい。また、仮想オブジェクトは画面上に表示しないことが望ましいが、表示するようにし

てもよい。

【0012】また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、前記エフェクト領域の画像を、テクスチャ座標をシフトさせながらテクセル補間方式で、前記エフェクト領域の形状の仮想オブジェクトにテクスチャマッピングすることを特徴とする。このようにすれば、テクセル補間方式を有効利用して、簡素な処理でエフェクト画像（画像エフェクト処理が施された画像）を生成できるようになる。

【0013】なお、テクセル補間方式とは、特に限定はされないが、テクセルの画像情報を補間してピクセルの画像情報を得る方式などであり、例えば、バイリニアフィルタ方式やトライリニアフィルタ方式などを考えることができる。

【0014】また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、前記エフェクト領域の画像の情報を、インデックスカラー・テクスチャマッピング用のルックアップテーブルのインデックス番号として設定し、該ルックアップテーブルを用いて、前記エフェクト領域の形状の仮想オブジェクトにインデックスカラー・テクスチャマッピングを行うことを特徴とする。このようにすれば、ゲームシステム（画像生成システム）が元々有しているインデックスカラー・テクスチャマッピングの機能を有効利用して、エフェクト画像を生成できる。従って、種々の画像エフェクト処理を、新たなハードウェアを追加することなく、高速に実行できるようになる。

【0015】なお、エフェクト領域の画像の情報は、例えば、描画領域（フレームバッファ、ワークバッファ等）に描画されている情報であり、色情報、 α 値又は奥行き値などを含む。また、また、エフェクト領域の画像の情報をルックアップテーブルのインデックス番号として設定することで得られる画像情報は、色情報に限定されない。

【0016】また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、前記エフェクト領域の画像に加工処理を施すことで得られた画像を、前記エフェクト領域の形状の仮想オブジェクトにテクスチャマッピングすることを特徴とする。この場合の加工処理としては、エフェクト領域の画像を何らかの形態で変換する処理であればよく、ピクセル入れ替え処理、ピクセル平均処理、モザイク処理、影生成処理など、種々の処理を考えることができる。

【0017】また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、前記画像エフェクト処理が、前記オブジェクトの画像に輪郭線の画像を付加するための処理であることを特徴とする。このようにすれば、オブジェクトの画像に対する輪郭線の画像の付加を簡素な処理で実現できるようになり、アニメや漫画に登場するキャラクタなどを表現するオブジェクトに最適な画像を

生成できるようになる。

【0018】

【発明の実施の形態】以下、本発明の好適な実施形態について図面を用いて説明する。

【0019】1. 構成

図1に、本実施形態のゲームシステム（画像生成システム）の機能ブロック図の一例を示す。なお同図において本実施形態は、少なくとも処理部100を含めばよく（或いは処理部100と記憶部170を含めばよく）、それ以外のブロックについては任意の構成要素とすることができる。

【0020】操作部160は、プレーヤが操作データを入力するためのものであり、その機能は、レバー、ボタン、マイク、或いは筐体などのハードウェアにより実現できる。

【0021】記憶部170は、処理部100や通信部196などのワーク領域となるもので、その機能はRAMなどのハードウェアにより実現できる。

【0022】情報記憶媒体（コンピュータにより使用可能な記憶媒体）180は、プログラムやデータなどの情報を格納するものであり、その機能は、光ディスク（CD、DVD）、光磁気ディスク（MO）、磁気ディスク、ハードディスク、磁気テープ、或いはメモリ（ROM）などのハードウェアにより実現できる。処理部100は、この情報記憶媒体180に格納される情報に基づいて本発明（本実施形態）の種々の処理を行う。即ち情報記憶媒体180には、本発明（本実施形態）の手段（特に処理部100に含まれるブロック）を実行するための情報（プログラム或いはデータ）が格納される。

【0023】なお、情報記憶媒体180に格納される情報の一部又は全部は、システムへの電源投入時等に記憶部170に転送されることになる。また情報記憶媒体180には、本発明の処理を行うためのプログラム、画像データ、音データ、表示物の形状データ、本発明の処理を指示するための情報、或いはその指示に従って処理を行うための情報などを含ませることができる。

【0024】表示部190は、本実施形態により生成された画像を出力するものであり、その機能は、CRT、LCD、或いはHMD（ヘッドマウントディスプレイ）などのハードウェアにより実現できる。

【0025】音出力部192は、本実施形態により生成された音を出力するものであり、その機能は、スピーカなどのハードウェアにより実現できる。

【0026】携帯型情報記憶装置194は、プレーヤの個人データやゲームのセーブデータなどが記憶されるものであり、この携帯型情報記憶装置194としては、メモリカードや携帯型ゲーム装置などを考えることができる。

【0027】通信部196は、外部（例えばホスト装置や他のゲームシステム）との間で通信を行うための各種

の制御を行うものであり、その機能は、各種プロセッサ、或いは通信用ASICなどのハードウェアや、プログラムなどにより実現できる。

【0028】なお本発明（本実施形態）の手段を実行するためのプログラム或いはデータは、ホスト装置（サーバー）が有する情報記憶媒体からネットワーク及び通信部196を介して情報記憶媒体180に配信するようにしてもよい。このようなホスト装置（サーバー）の情報記憶媒体の使用も本発明の範囲内に含まれる。

【0029】処理部100（プロセッサ）は、操作部160からの操作データやプログラムなどに基づいて、ゲーム処理、画像生成処理、或いは音生成処理などの各種の処理を行う。この処理部100の機能は、各種プロセッサ（CPU、DSP等）又はASIC（ゲートアレイ等）などのハードウェアや、所与のプログラム（ゲームプログラム）により実現できる。

【0030】ここで、処理部100が行うゲーム処理としては、コイン（代価）の受け付け処理、各種モードの設定処理、ゲームの進行処理、選択画面の設定処理、オブジェクト（1又は複数のプリミティブ面）の位置や回転角度（X、Y又はZ軸回り回転角度）を求める処理、オブジェクトを動作させる処理（モーション処理）、視点の位置（仮想カメラの位置）や視線角度（仮想カメラの回転角度）を求める処理、マップオブジェクトなどのオブジェクトをオブジェクト空間へ配置する処理、ヒットチェック処理、ゲーム結果（成果、成績）を演算する処理、複数のプレーヤが共通のゲーム空間でプレイするための処理、或いはゲームオーバー処理などを考えることができる。

【0031】また、処理部100は、上記のゲーム処理結果に基づいて例えばオブジェクト空間内において所与の視点（仮想カメラ）から見える画像を生成し、表示部190に出力する。

【0032】更に、処理部100は、上記のゲーム処理結果に基づいて各種の音処理を行い、BGM、効果音、又は音声などの音を生成し、音出力部192に出力する。

【0033】なお、処理部100の機能は、その全てをハードウェアにより実現してもよいし、その全てをプログラムにより実現してもよい。或いは、ハードウェアとプログラムの両方により実現してもよい。

【0034】処理部100は、ジオメトリ処理部110、エフェクト領域設定部112（仮想オブジェクト生成部）、インデックス番号設定部114、描画部120を含む。

【0035】ここでジオメトリ処理部110は、座標変換、クリッピング処理、透視変換、或いは光源計算などの種々のジオメトリ処理（3次元演算）を、オブジェクトに対して行う。ジオメトリ処理により得られた描画データ（頂点に付与される位置座標、テクスチャ座標、色

（輝度）データ、法線ベクトル或いは α 値等）は、記憶部170の主記憶部172に格納されて、保存される。

【0036】エフェクト領域設定部112（仮想オブジェクト生成部）は、画像エフェクトを施す領域であるエフェクト領域を設定するための処理を行う。より具体的には、エフェクト領域の形状の仮想オブジェクト（狭義には仮想ポリゴン）を生成する。

【0037】本実施形態では、このエフェクト領域（仮想オブジェクト）が、透視変換後（スクリーン座標系への変換後）のオブジェクトの画像の全部又は一部を内包し、透視変換後のオブジェクトの大きさに応じてその大きさが変化するようにになっている。即ち、画面内でのオブジェクトの占有面積に応じてその大きさが変化するようにになっている。そして、画像エフェクト処理は、このエフェクト領域の画像に対して行われるようになるため、無駄の無い画像エフェクト処理を実現できる。

【0038】インデックス番号設定部114は、エフェクト領域の画像の情報（元画像情報）を、LUT（ルックアップテーブル）記憶部178に記憶されるインデックスカラー・テクスチャマッピング用のLUTのインデックス番号として設定するための処理を行う。ここで、インデックス番号として設定されるエフェクト領域の画像情報としては、例えば、色情報（RGB、YUV等）、 α 値（各ピクセルに関連づけられて記憶される情報であり色情報以外のプラスアルファの情報）、奥行き値（Z値）等、種々の情報を考えることができる。

【0039】描画部120（オブジェクト・輪郭線描画部）は、ジオメトリ処理後のオブジェクト（モデル）や、オブジェクトの輪郭線を、描画領域174（フレームバッファ、ワークバッファ等のピクセル単位で画像情報を記憶できる領域）に描画するための処理を行うものであり、テクスチャマッピング部130（画像エフェクト処理部）、合成部132、陰面消去部134を含む。

【0040】テクスチャマッピング部130は、テクスチャ記憶部176に記憶されるテクスチャをオブジェクトにマッピングするための処理を行う。

【0041】より具体的には、テクスチャマッピング部130は、エフェクト領域の画像を用いた、エフェクト領域の形状の仮想オブジェクトへのテクスチャマッピング処理を行う。例えば、エフェクト領域の画像を、テクスチャ座標を例えば1テクセル（ピクセル）よりも小さい値だけシフトさせながらテクセル補間方式（狭義にはバイニアフィルタ）で、エフェクト領域の形状の仮想オブジェクトにテクスチャマッピングする。或いは、エフェクト領域の画像情報がインデックス番号として設定されたLUT（ルックアップテーブル）を用いて、仮想オブジェクトに対するテクスチャマッピングを行う。或いは、エフェクト領域の画像に加工処理（ピクセル入れ替え処理、ピクセル平均処理等）を施すことで得られた画像を、仮想オブジェクトにテクスチャマッピングする。

【0042】合成部132は、 α 値を用いたマスク合成処理や半透明処理を行う。なお、 α 値（A値）は、各ピクセルに関連づけられて記憶される情報であり、例えば色情報以外のプラスアルファの情報である。 α 値は、マスク情報、透明度（不透明度、半透明度と等価）、パンプ情報などとして使用できる。

【0043】陰面消去部134は、Z値（奥行き値）が格納されるZバッファ（Zプレーン）を用いて、Zバッファ法のアルゴリズムにしたがった陰面消去を行う。但し、視点からの距離に応じてプリミティブ面をソーティングし、視点から遠い順にプリミティブ面を描画する奥行きソート法（Zソート法）などにより陰面消去を行ってもよい。

【0044】なお、本実施形態のゲームシステムは、1人のプレーヤのみがプレイできるシングルプレーヤモード専用のシステムにしてもよいし、このようなシングルプレーヤモードのみならず、複数のプレーヤがプレイできるマルチプレーヤモードも備えるシステムにしてもよい。

【0045】また複数のプレーヤがプレイする場合に、これらの複数のプレーヤに提供するゲーム画像やゲーム音を、1つの端末を用いて生成してもよいし、ネットワーク（伝送ライン、通信回線）などで接続された複数の端末を用いて生成してもよい。

【0046】2. 本実施形態の特徴

2. 1 エフェクト領域の設定

いわゆる3次元ゲームの画像を生成できるこれまでのゲームシステムでは、キャラクタなどを表すオブジェクトは複数のポリゴンや自由曲面（広義にはプリミティブ面）により構成される。そして、このポリゴンにより構成されたオブジェクト（モデル）をオブジェクト空間内に配置し、ジオメトリ処理（3次元演算）を行って、オブジェクト空間内の所与の視点から見える画像を生成する。これにより、あたかも現実世界の風景のようにリアルで実写に近い画像を生成できるようになる。

【0047】一方、アニメや漫画などの分野では、実写のようなリアルな画像で人を魅了するのではなく、アニメ特有のセル画風の画像で人を魅了している。

【0048】従って、これまでのゲームシステムで生成されるゲーム画像は、リアリティを好む人の情感に訴えることはできるが、アニメや漫画などを好む人の情感には今一つ訴えることができないという課題がある。

【0049】このような課題を解決するために、本出願人は、セル画風の画像をリアルタイムに生成できるゲームシステムの開発を行っている。

【0050】図2に、本実施形態のゲームシステムにより生成されるゲーム画像の例を示す。図2に示すように本実施形態によれば、ポリゴンで構成されたオブジェクトOBの輪郭に沿って太い輪郭線が描かれており、漫画やアニメなどで一般の人が親しんでいるセル画風の画像

の生成に成功している。

【0051】さて、図2に示すような画像を生成するためには、オブジェクトに輪郭線を描画するための種々の画像エフェクト処理が必要になる。そして、このような画像エフェクト処理は、フレームバッファに描画された2次元の画面全体に対して行われることになる。

【0052】ところが、このように画面全体に対して画像エフェクト処理を施すと、処理に無駄があり、効率的な画像エフェクト処理を実現できないという課題があることが判明した。

【0053】即ち、3次元ゲームにおいては、プレーヤの操作入力などに応じてプレーヤの視点は任意の位置に移動し、キャラクタを表現するオブジェクトも任意の位置に移動する。従って、プレーヤの視点とオブジェクトとの間の距離も様々なに変化し、視点とオブジェクトとの距離に応じて、透視変換後のオブジェクトの大きさ（画面内でのオブジェクトの占有面積）も様々なに変化ようになる。

【0054】例えば図3（A）に示すように、視点とオブジェクトOBとの距離が近い場合には、透視変換後のオブジェクトOBの大きさは大きくなる（画面での占有面積が大きくなる）。

【0055】一方、図3（B）に示すように、視点とオブジェクトOBとの距離が近い場合には、透視変換後のオブジェクトOBの大きさは小さくなる（画面での占有面積が小さくなる）。

【0056】図3（A）に示すように画面内でのオブジェクトOBの占有面積が大きい場合には、輪郭線を付加するための画像エフェクト処理を画面全体に対して施しても、それほど不効率ではない。

【0057】ところが、図3（B）に示すように画面内でのオブジェクトOBの占有面積が小さい場合には、輪郭線を付加するための画像エフェクト処理を画面全体に対して施すと、処理に無駄が多く、画像エフェクト処理の負荷が不必要に重くなってしまう。

【0058】そこで本実施形態では、透視変換後のオブジェクトOBの画像を内包し、透視変換後のOBの大きさに応じてその大きさが変化する図3（A）、（B）に示すエフェクト領域EAを設定する。そして、このエフェクト領域EA内の画像（元画像）に対して画像エフェクト処理を施すようにしている。

【0059】このようにすれば、視点とオブジェクトOBとの距離が変化し、画面内でのOBの占有面積が変化した場合に、それに応じてエフェクト領域EAの大きさも変化し、画像エフェクト処理を施す範囲も変化するようになる。例えば、視点とOBとの距離が遠くなると、エフェクト領域EAの大きさが小さくなり、画像エフェクト処理を施す範囲も狭い範囲で済むようになる。従って、画像エフェクト処理の効率化を図れ、画像エフェクト処理の負荷を最適化できるようになる。

【0060】なお、オブジェクトがパーツオブジェクトにより構成される場合には、各パーツオブジェクト毎にエフェクト領域を設定するようにしてもよい。また、オブジェクトの一部（例えば目、口）にだけ画像エフェクト処理を施す場合には、エフェクト領域EAは、透視変換後のオブジェクトの一部にだけ設定するようにしてもよい。

【0061】さて、エフェクト領域EAの設定手法としては種々の手法を考えることができる。

【0062】第1の手法では、図4（A）、（B）に示すように、透視変換後のオブジェクトOBの頂点（広義には自由曲面の制御点なども含む定義点） $VX1 \sim VX6$ 等の座標に基づいて、エフェクト領域EAを設定する。

【0063】より具体的には、オブジェクトOBの頂点のX、Y座標の最小値XMIN、YMIN、最大値XMAX、YMAXに基づいて、エフェクト領域EAの頂点 $VX1$ （XMIN、YMIN）、 $VX2$ （XMIN、YMAX）、 $VX3$ （XMAX、YMAX）、 $VX4$ （XMAX、YMIN）を求め、エフェクト領域EAを特定する。

【0064】この第1の手法によれば、エフェクト領域EAの大きさを、透視変換後のオブジェクトOBの大きさに応じて最適に小さくできるため、画像エフェクト処理の負荷が軽くなるという利点がある。一方、オブジェクトOBの頂点が多い場合には、それらの頂点のX、Y座標の最小値、最大値を求める処理の負荷が重くなるという欠点がある。

【0065】第2の手法では、図5（A）に示すように、オブジェクトOBを内包する簡易オブジェクトSOB（バウンディングボックス、バウンディングボリューム）を利用する。この簡易オブジェクトSOBは、オブジェクトOBのヒットチェック処理等のために使用されるものである。本実施形態ではこの簡易オブジェクトSOBを有効利用し、透視変換後の簡易オブジェクトSOBの頂点（広義には定義点）の座標に基づいてエフェクト領域EAを設定する。

【0066】より具体的には図5（B）に示すように、透視変換後の簡易オブジェクトSOBの頂点 $VSX1 \sim VSX8$ のX、Y座標の最小値、最大値に基づいて、エフェクト領域EAの頂点 $VX1 \sim VX4$ を求め、エフェクト領域EAを特定する。

【0067】この第2の手法によれば、オブジェクトOBに比べて頂点数が少ない簡易オブジェクトSOBを用いるため、頂点のX、Y座標の最小値、最大値を求める処理の負荷が軽くなるという利点がある。一方、第1の手法に比べて、エフェクト領域EAの大きさが大きくなってしまいうため、画像エフェクト処理の負荷が第1の手法に比べて重くなるという欠点がある。

【0068】なお、後述するようにオブジェクトの輪郭

の外側領域に輪郭線がはみ出す場合には、はみ出した輪郭線が適正に描画されるように、画像エフェクト領域EAの大きさを、図4（A）、（B）、図5（B）に示す大きさよりも上下左右方向に若干だけ大きくすることが望ましい（例えば1ピクセル分だけ大きくする）。

【0069】また、透視変換後のオブジェクトの大きさに応じてエフェクト領域の大きさを変化させる手法は、上述の第1、第2の手法に限定されない。例えば、視点とオブジェクトとの間の距離に応じてエフェクト領域の大きさを適宜変化させるようにしてもよい。

【0070】2. 2 テクスチャマッピングを利用した画像エフェクト処理

さて、以上のように設定されたエフェクト領域の画像に画像エフェクト処理を施す手法としては種々の手法を考えることができる。そして本実施形態では特に、エフェクト領域の画像を用いたテクスチャマッピングにより、エフェクト領域の画像に対する画像エフェクト処理を実現している。より具体的には、エフェクト領域の形状の仮想ポリゴン（広義には仮想オブジェクト）を生成し、この仮想ポリゴンに対するテクスチャマッピングにより、画像エフェクト処理を実現している。

【0071】このようなテクスチャマッピングを利用した画像エフェクト処理としては、例えば以下に説明するような手法を考えることができる。

【0072】2. 2. 1 インデックスカラー・テクスチャマッピングの利用

インデックスカラーテクスチャマッピングでは、テクスチャ記憶部の使用記憶容量を節約するために、図6のA1に示すように、実際の色情報（RGB）ではなくインデックス番号が、テクスチャの各テクセルに関連づけて記憶される。また、図6のA2に示すように、インデックスカラー・テクスチャマッピング用のルックアップテーブルLUT（カラーパレット）には、インデックス番号により指定される色情報が記憶される。そして、オブジェクトに対してテクスチャマッピングを行う際には、テクスチャの各テクセルのインデックス番号に基づいてLUTを参照し、対応する色情報をLUTから読み出し、読み出された色情報をフレームバッファに描画する。

【0073】このようなインデックスカラーモードのテクスチャマッピングでは、LUTを用いない通常モードのテクスチャマッピングに比べて、使用できる色数は少なくなる（例えば16色）。しかしながら、テクスチャ記憶部に実際の色情報（例えば16ビットの色情報）を記憶する必要がなくなるため、テクスチャ記憶部の使用記憶容量を大幅に節約できる。

【0074】本実施形態は、このようなインデックスカラー・テクスチャマッピングを通常とは異なる形態で利用する。

【0075】即ち、まず図7のB1に示すように、エフ

エフェクト領域の各ピクセルの画像情報（例えば色情報）を、ルックアップテーブルLUTのインデックス番号として設定する（インデックス番号とみなす）。そしてB2に示すように、エフェクト領域の画像情報がインデックス番号として設定されたLUTを用いて、仮想ポリゴン（エフェクト領域の形状のポリゴン）に対してインデックスカラー・テクスチャマッピングを行い、エフェクト領域の画像に対して画像エフェクト処理を施す。例えば図7のB3に示すようなLUTを用いれば、図8

(A)に示すようなガンマ補正をエフェクト領域の画像に対して施すことができるようになる。

【0076】通常、ガンマ補正を実現するためには専用のハードウェアであるガンマ補正回路が必要になる。しかしながら、このようなガンマ補正回路を別途設けると、ゲームシステムの高コスト化を招く。特に、家庭用のゲームシステムでは、製品の普及化を図るために低コスト化が厳しく要求されており、このようなガンマ補正回路が設けられていないのが一般的である。従って、このような家庭用のゲームシステムでは、ハードウェアによるガンマ補正は実現不可能となる。

【0077】また、ガンマ補正をソフトウェア処理によ

$$Y = Y20 + (X/255) \times (Y21 - Y20) \quad (1)$$

但し、

$$Y20 = Y10 + (X/255) \times (Y11 - Y10)$$

$$Y21 = Y11 + (X/255) \times (Y12 - Y11)$$

$$Y10 = Y0 + (X/255) \times (Y1 - Y0)$$

$$Y11 = Y1 + (X/255) \times (Y2 - Y1)$$

$$Y12 = Y2 + (X/255) \times (Y3 - Y2)$$

である。

【0082】上式(1)の入力値Xにインデックス番号を設定し、出力値Yに各色成分の出力ROUT、GOUT、BOUTを設定することで、図8(B)に示すようなガンマ補正用LUTを作成する。そして、作成されたLUTをVRAMに転送し、このLUTを用いて図7で説明したインデックスカラー・テクスチャマッピングを行うことで、エフェクト領域の画像にガンマ補正を施した画像を得ることができる。

【0083】なお図7の手法によれば、ガンマ補正以外にも種々の画像エフェクト処理（ビデオフィルタ）を実現できる。

【0084】例えば図9(A)はネガポジ反転の変換特性例であり、図9(B)はネガポジ反転用に作成されたLUTの例である。

【0085】また図10(A)、(B)、(C)は、各々、ポストリザイゼーション、ソラリゼーション、2値化などの画像エフェクト処理の変換特性例である。図7の手法によれば、これ以外にも、モノトーンフィルタやセピアフィルタなどの種々の画像エフェクト処理を実現できる。

【0086】なお、ガンマ補正などの画像エフェクト処

り実現する手法も考えることができるが、処理速度、処理負荷の観点からは現実的ではない。

【0078】これに対して図7の手法では、インデックスカラー・テクスチャマッピングを有効利用してガンマ補正などの画像エフェクト処理を実現しており、このインデックスカラー・テクスチャマッピングは、専用のハードウェアである描画プロセッサ（描画部）により高速に実行される。従って、ソフトウェア処理で実現する手法に比べて高速にガンマ補正を実行できるという利点がある。

【0079】また図7の手法では、インデックスカラー・テクスチャマッピングは描画プロセッサが元々持っているハードウェアにより実行されるため、ガンマ補正回路を新たに付加する必要がなく、ゲームシステムが高コスト化してしまう事態を防止できる。また、ガンマ補正回路を元々有していない家庭用ゲームシステムにおいても、ハードウェアによる高速なガンマ補正を実現できるようになる。

【0080】さて、ガンマ補正における入力値Xと出力値Yの関係式は、例えば下式のようなになる。

【0081】

理では、1つの入力値(RIN、GIN又はBIN)に対して1つの値(ROUT、GOUT又はBOUT)が出力されるようなLUTが必要になる。

【0087】ところが、図6に示すインデックスカラー・テクスチャマッピング用のLUTは、元々、ガンマ補正用に設計されたものではないため、1つ入力値（インデックス番号）に対して、複数の値（例えばROUT、GOUT、及びBOUT）が出力されてしまう。従って、このLUTの不整合を解決しなければならないという課題がある。

【0088】このような課題を解決する1つの手法として、エフェクト領域の画像の中の1つの色成分をLUTのインデックス番号として設定する場合において、変換により得られる他の色成分をマスクするという手法を考えることができる。

【0089】より具体的には、図11のD1に示すように、エフェクト領域の画像のRプレーンの値をインデックス番号に設定して、LUTを用いたテクスチャマッピングを行うと、R(ROUT)、G(GOUT)、B(BOUT)という3つプレーンの値が出力される。そして、この場合には、D2に示すように、出力されたRプレーンの値のみを用い、Gプレーン、Bプレーンの値についてはマスクするようにする。

【0090】また、図11のD3に示すように、エフェクト領域の画像のGプレーンの値をインデックス番号に設定してテクスチャマッピングを行った場合には、D4に示すように、出力されたGプレーンの値のみを用い、Rプレーン、Bプレーンの値についてはマスクするよう

にする。

【0091】また、図11のD5に示すように、エフェクト領域の画像のBプレーンの値をインデックス番号に設定してテクスチャマッピングを行った場合には、出力されたBプレーンの値のみを用い、Rプレーン、Gプレーンの値についてはマスクするようにする。

【0092】以上のようにすることで、元々はその画像エフェクト処理のために専用には設計されていないインデックスカラー・テクスチャマッピングのLUTを用いながらも、少ない処理負荷でその画像エフェクト処理を実現できるようになる。

【0093】なお、インデックスカラー・テクスチャマッピング用のLUTに基づき出力される情報は、色情報には限定されず、 α 値であってもよい。この場合には、例えば、Rプレーン（又はG、B）の値をインデックス番号として設定して、LUTを用いたテクスチャマッピングを行い、 α プレーンを生成する。そして、生成された α プレーンを用いて、マスク合成や半透明合成などを行うようにする。

【0094】また、LUTのインデックス番号として設定される画像情報は色情報に限定されない。即ち、描画領域（VRAM）上にあり、LUTのインデックス番号として設定できる画像情報であればよく、Z値（奥行き値）をLUTのインデックス番号として設定するようにしてもよい。この場合には、例えば、Z値をインデックス番号に設定してインデックスカラー・テクスチャマ

$$\begin{aligned} CP = & (1-\beta) \times (1-\gamma) \times CA + \beta \times (1-\gamma) \times CB \\ & + (1-\beta) \times \gamma \times CC + \beta \times \gamma \times CD \end{aligned} \quad (2)$$

本実施形態では、このようにバイリニアフィルタ方式では色が自動的に補間されることに着目して、ぼかし画像を生成している。

【0102】より具体的には図13のF1に示すように、エフェクト領域の画像（マッピング画像）をテクスチャとして設定する。そして図13のF2に示すように、このテクスチャを仮想ポリゴン（広義には仮想オブジェクト）にバイリニアフィルタ方式でマッピングする際に、仮想ポリゴンの頂点に与えるテクスチャ座標を、例えば（0.5、0.5）だけ右下方向にシフト（ずらす、移動）させる。このようにすることで、バイリニアフィルタ方式の補間機能により自動的に、エフェクト領域の画像の各ピクセルの色が、そのピクセルの周囲のピクセルににじむようになる。これにより、ぼかし画像の生成が可能になる。

【0103】なお、エフェクト領域の頂点VVX1、VVX2、VVX3、VVX4の座標が（X、Y）＝（X0、Y0）、（X0、Y1）、（X1、Y1）、（X1、Y0）であったとする。この場合に、エフェクト領域の形状の仮想ポリゴンの頂点VVX1、VVX2、VVX3、VVX4に与えるテクスチャ座標（U、V）を、各々、（X0、Y0）、（X0、Y1）、（X1、Y1）、

（X1、Y0）に設定すれば、ピクセルの位置とテクセルの位置とがずれずに一致する。従って、画像はぼけない。

【0095】2. 2. 2 バイリニアフィルタ方式のテクスチャマッピングの利用バイリニアフィルタ方式（広義にはテクセル補間方式）のテクスチャマッピングを有効利用することで、ぼかし画像の生成（或いは後述するような輪郭線画像の生成）が可能になる。

【0096】即ち、テクスチャマッピングにおいてはピクセルの位置とテクセルの位置がずれる場合がある。

【0097】この場合に、図12に示すように、ポイントサンプリング方式では、ピクセル（サンプリング点）Pの色CP（広義には画像情報）は、Pに最も距離に近いテクセルTAの色CAになる。

【0098】一方、バイリニアフィルタ方式では、Pの色CPは、Pの周りのテクセルTA、TB、TC、TDの色CA、CB、CC、CDを補間した色になる。

【0099】より具体的には、TA～TDの座標とPの座標とに基づき、X軸方向の座標比 β ： $1-\beta$ （ $0 \leq \beta \leq 1$ ）と、Y軸方向の座標比 γ ： $1-\gamma$ （ $0 \leq \gamma \leq 1$ ）を求める。

【0100】この場合に、Pの色CP（バイリニアフィルタ方式での出力色）は、下式のようなになる。

【0101】

（X1、Y0）に設定すれば、ピクセルの位置とテクセルの位置とがずれずに一致する。従って、画像はぼけない。

【0104】これに対して、仮想ポリゴンの頂点VVX1、VVX2、VVX3、VVX4に与えるテクスチャ座標（U、V）を、各々、（X0+0.5、Y0+0.5）、（X0+0.5、Y1+0.5）、（X1+0.5、Y1+0.5）、（X1+0.5、Y0+0.5）に設定すれば、ピクセルの位置とテクセルの位置とがずれるようになる。従って、バイリニアフィルタ方式の補間機能により、色の補間が行われ、画像がぼけて見えるようになる。

【0105】次に、バイリニアフィルタ方式の補間機能によりぼかし画像が生成される原理について説明する。

【0106】例えば図14（A）に示すように、テクスチャ座標を0.5テクセルだけ右下方向にシフトさせて、バイリニアフィルタ方式のテクスチャマッピングを行ったとする。この場合には、上式（2）において $\beta = \gamma = 1/2$ になるため、テクセルT44、T45、T54、T55の色をC44、C45、C54、C55とすると、ピクセルP44の色CP44は下式のようなになる。

【0107】

$$CP44 = (C44 + C45 + C54 + C55) / 4 \quad (3)$$

以上から明らかなように、図14(A)に示す変換により、テクセルT44の色C44(変換前の元画像のピクセルP44の元の色に相当)は、周りのピクセルP33、P34、P43、P44に対して1/4ずつしみ出すことになる。

【0108】そして、その後に図14(B)に示すように、図14(A)で得られた画像をテクスチャとして、テクスチャ座標を0.5テクセルだけ左上方向にシフトさせてバイリニアフィルタ方式でテクスチャマッピングを行ったとする。この場合には、図14(A)のピクセルP33、P34、P43、P44が、図14(B)のテクセルT33、T34、T43、T44に対応するようになる。そして、図14(A)でP33、P34、P43、P44(T33、T34、T43、T44)に対して1/4ずつしみ出した色C44が、更に1/4倍されて周りの4つのピクセルに対してしみ出すことになる。即ち、結局、元のT44の色C44が $1/4 \times 1/4 = 1/16$ ずつ周りにしみ出すことになる。

【0109】従って、図14(A)、(B)の変換により、ピクセルP33、P34、P35には、各々、色C44(フレームバッファに描かれた元画像のピクセルP44の元の色に相当)が $1/16$ 、 $2/16$ 、 $1/16$ ずつしみ出すことになる。また、ピクセルP43、P44、P45には、各々、色C44が $2/16$ 、 $4/16$ 、 $2/16$ ずつしみ出し、ピクセルP53、P54、P55には、各々、色C44が $1/16$ 、 $2/16$ 、 $1/16$ ずつしみ出すことになる。

【0110】従って、図14(A)、(B)の変換により、結局、図15(A)に示すような平面フィルタが元画像に対して施されるようになる。この平面フィルタによれば、エフェクト領域の画像の各ピクセルの色がその周りに均一に広がるようになり、理想的なぼかし画像を生成できる。

【0111】また、図14(A)、(B)の変換のセットを2回行えば、図15(B)に示すような平面フィルタがエフェクト領域の画像に対して施されるようになる。この平面フィルタによれば、図15(A)よりも更に理想的なぼかし画像を生成できる。

【0112】なお、後述するように、以上のようなバイリニアフィルタ方式のテクスチャマッピングを利用することで、ぼかし画像のみならず、オブジェクトの輪郭線の画像も生成できるようになる。

【0113】2.2.3 加工処理画像のテクスチャマッピング

さて、テクスチャマッピングを利用した画像エフェクト処理としては、上記の2.2.1や2.2.2で説明した手法以外にも、種々の手法を考えることができる。

【0114】例えば、エフェクト領域の画像に加工処理

を施すことで得られた画像を、エフェクト領域の形状の仮想ポリゴン(仮想オブジェクト)にテクスチャマッピングするようにしてもよい。

【0115】より具体的には、エフェクト領域の画像に対して、図16(A)、(B)、(C)に示すようなピクセル入れ替え処理を施し、ピクセル入れ替え後の画像を仮想ポリゴンにマッピングする。例えば図16(B)では、RとHのピクセルの色情報が入れ替わっており、図16(C)では、JとQのピクセルの色情報が入れ替わっている。

【0116】図16(A)、(B)、(C)に示すようなピクセル入れ替え処理を行うことで、擬似的に光が屈折したかのように見える表現が可能になる。

【0117】なお、エフェクト領域の画像に施す加工処理としては、ピクセル入れ替え処理以外にも、ピクセル平均処理、モザイク処理、輝度変換処理等の種々の加工処理を考えることができる。

【0118】また、過去のフレームでのエフェクト領域の画像を記憶部に保持しておき、当該フレームでのエフェクト領域の画像と過去のフレームでのエフェクト領域の画像とを合成する加工処理を行ってもよい。

【0119】2.3 輪郭線画像

次に、オブジェクトの輪郭線の画像を生成する手法について説明する。

【0120】本実施形態では上述の2.2.2で説明したバイリニアフィルタ方式のテクスチャマッピングを利用して輪郭線(edge line)の画像を生成している。

【0121】2.3.1 マッピング画像の生成

より具体的には、まず前処理として、ワークバッファ(エフェクトバッファ)を輪郭線の画像情報(RL、GL、BL、 α L)で初期化する。即ち、各ピクセルの色を輪郭線の色(RL、GL、BL)に設定すると共に、 α 値を α L($=0$)に設定する。

【0122】次に、輪郭線付加の対象となるオブジェクトOBをワークバッファに描画する。この場合に、オブジェクトOBの頂点の α 値(A値)を α J(>0)に設定しておく。なお、説明を簡単にするために、オブジェクトOBの色は単色の(RJ、GJ、BJ)であるとする。

【0123】以上のようにすることで、ワークバッファには図17に示すような画像(マッピング画像)が描画されることになる。

【0124】即ち、オブジェクトOBの輪郭EDの内側領域はOBの色(RJ、GJ、BJ)に設定され、輪郭EDの外側領域は輪郭線の色(RL、GL、BL)に設定される。また、内側領域の α 値は α J(>0)に設定され、外側領域の α 値は α L($=0$)に設定される。

【0125】なお、図17に示す α 値の設定は一例であ

り、少なくとも外側領域の α 値と内側領域の α 値が異なった値に設定されていればよい。

【0126】2. 3. 2 バイリニアフィルタ方式でのテクスチャマッピング

次に、図17に示すワークバッファ上のマッピング画像をフレームバッファ上の同じ位置に描画する。

【0127】より具体的には図18に示すように、ワークバッファ上のマッピング画像を仮想ポリゴン（広義には仮想オブジェクト）にマッピングしながら、その仮想ポリゴンをフレームバッファに描画する。

【0128】この場合に、図3（A）～図5（B）で説明したように、オブジェクトを内包し且つ透視変換後のオブジェクトの大きさに応じてその大きさが変化するような形状の仮想ポリゴン（エフェクト領域）を生成し、その仮想ポリゴンに対してワークバッファ上のマッピング画像をマッピングする。

【0129】また、このテクスチャマッピングの際に、①バイリニアフィルタ（テクセル補間）方式を選択し、図13で説明したようにテクスチャ座標を例えば+0.5ピクセル（テクセル）だけシフトさせ（ずらす）、

ソース α テスト（書き込み元であるマッピング画像に対する α テスト）で $\alpha > 0$ のピクセル（ α が0でないピクセル）だけを合格にする。

【0130】これにより、オブジェクトOBに対して輪郭線EDLが付与された図19に示すような画像が生成される。

【0131】即ち、バイリニアフィルタを選択すると共にテクスチャ座標をシフトさせることで、ワークバッファ上のマッピング画像の各ピクセルに設定される色及び α 値（A値）の補間が行われるようになる。

【0132】例えば描画ピクセルの位置が（X、Y）で、テクスチャ座標U、Vを+0.5ピクセル（テクセル）だけシフトしたとすると、テクスチャサンプリング時には、（X、Y）、（X+1、Y）、（X、Y+1）、（X+1、Y+1）の4つのピクセルの位置にある色及び α 値が参照されることになる。この結果、最終的に描画されるテクセルの色及び α 値は、上記の4つのピクセルの色及び α 値の平均となる。例えば図20に示すように、テクスチャ座標を右下方向にシフトさせた場合には、ピクセルB、C、DのRGB、 α 値が1/4ずつピクセルAに対してしみ出すことになる。これを式で表すと下式のようになる。

【0133】

$$R = (RA + RB + RC + RD) / 4$$

$$G = (GA + GB + GC + GD) / 4$$

$$B = (BA + BB + BC + BD) / 4$$

$$\alpha = (\alpha A + \alpha B + \alpha C + \alpha D) / 4 \quad (4)$$

上式において、R、G、B、 α は、補間により得られる色及び α 値（補間後のピクセルAの色及び α 値）であ

る。また、（RA、GA、BA、 αA ）、（RB、GB、BB、 αB ）、（RC、GC、BC、 αC ）（RD、GD、BD、 αD ）は、各々、補間前のピクセルA、B、C、Dの色及び α 値である。

【0134】以上のように、テクスチャ座標を右下方向に0.5ピクセルだけシフトさせると、ピクセルB、C、DのRGB、 α 値が1/4ずつピクセルAに対してしみ出す。従って、図19のH1に示す部分では、図21（A）に示すように、輪郭EDの内側領域においてオブジェクトOBの色（RJ、GJ、BJ）と輪郭線の色（RL、GL、BL）とが合成され、輪郭EDの内側領域に輪郭線EDLの画像が生成されるようになる。

【0135】なお、この輪郭線の色（RL、GL、BL）は、図17に示すようにワークバッファにおいて輪郭EDの外側領域に設定された色である。

【0136】一方、図19のH2に示す部分では、図21（B）に示すように、輪郭EDの外側領域においてオブジェクトOBの色（RJ、GJ、BJ）と輪郭線の色（RL、GL、BL）とが合成され、輪郭EDの外側領域に輪郭線EDLの画像が生成されるようになる。これによって、結局、図19に示すように、オブジェクトOBの輪郭EDに沿って輪郭線EDLが描画されるようになる。

【0137】なお、本実施形態においては、色のみならず α 値もバイリニアフィルタによる補間処理の対象になる。従って、輪郭線EDLに設定される α 値も、オブジェクトOBの αJ と輪郭線の αL とを合成した α 値になる。

【0138】以上のことを更に詳しく説明すると以下のようになる。

【0139】（I）参照される周りのピクセルが全て輪郭線色になっているピクセル

図21（A）のI1や図21（B）のI2に示すピクセルでは、参照される周りのピクセル（テクセル）のRGB及び α 値は全て（RL、GL、BL、 αL ）になっている。従って、上式（4）において（RA、GA、BA、 αA ）＝（RB、GB、BB、 αB ）＝（RC、GC、BC、 αC ）＝（RD、GD、BD、 αD ）＝（RL、GL、BL、 αL ）となるため、補間後のRGB及び α 値は（R、G、B、 α ）＝（RL、GL、BL、 αL ）になる。

【0140】この時、 $\alpha = \alpha L = 0$ に設定されているため（図17参照）、 $\alpha > 0$ のピクセルのみを合格にするソース α テストが不合格となり、これらのピクセルについては描画が禁止される。

【0141】（II）輪郭線色のピクセルとオブジェクト色のピクセルの双方が参照されるピクセル

図21（A）のI3や図21（B）のI4に示すような、オブジェクトOBの輪郭ED付近のピクセルでは、その周りに、（RL、GL、BL、 αL ）のピクセル

(テクセル)と(RJ、GJ、BJ、αJ)のピクセル(テクセル)が混在する。この場合、(RJ、GJ、BJ、αJ)のピクセルの数をK個とすると、上式(4)により補間後の(R、G、B、α)は下式のようになる。

【0142】

$$\begin{aligned} R &= [RJ \times K + RL \times (4 - K)] / 4 \\ G &= [GJ \times K + GL \times (4 - K)] / 4 \\ B &= [BJ \times K + BL \times (4 - K)] / 4 \\ \alpha &= [\alpha J \times K + \alpha L \times (4 - K)] / 4 \end{aligned} \quad (5)$$

上式(5)のように、補間後のピクセルの色は、オブジェクトOBの色(RJ、GJ、BJ、αJ)と輪郭線の色(RL、GL、BL、αL)とが混合された色になる。

【0143】また、αJ>0、1≤K≤3、αL=0であるため、0<α<αJとなり、α>0のピクセルのみを合格にするソースαテストが合格となり、これらのピクセルについては必ず描画されるようになる。

【0144】(III)参照される周りのピクセルが全てオブジェクト色になっているピクセル図21(A)のI5や図21(B)のI6に示すピクセルでは、参照される周りのピクセル(テクセル)のRGB及びα値は全て(RJ、GJ、BJ、αJ)になっている。従って、上式(4)において(RA、GA、BA、αA)=(RB、GB、BB、αB)=(RC、GC、BC、αC)=(RD、GD、BD、αD)=(RJ、GJ、BJ、αJ)となるため、補間後のRGB及びα値は(R、G、B、α)=(RJ、GJ、BJ、αJ)になる。

【0145】この時、α=αJ>0に設定されているため(図17参照)、α>0のピクセルのみを合格にするソースαテストが合格となり、これらのピクセルについてはオブジェクトの色がそのまま描画されるようになる。

【0146】以上のようにしてフレームバッファに図19に示すような画像を描画できるようになる。

【0147】なお、図19では、オブジェクトOBの輪郭EDの付近のみならずOBの全体に対してバイリニアフィルタによる補間処理が施されてしまい、オブジェクトOBの全体がぼけた画像になってしまう。そこで本実施形態では、次のような手法を採用して、オブジェクトOBの全体がぼけた画像になってしまう問題を解消している。

【0148】2. 3. 3 フレームバッファへのオブジェクトの描画

前述の2. 3. 2と同様に、図17に示すワークバッファのマッピング画像をフレームバッファ上の同じ位置に描画する。具体的には、ワークバッファ上のマッピング画像を仮想ポリゴンにマッピングしながら、その仮想ポリゴンをフレームバッファに描画する。

【0149】この場合に、図3(A)～図5(B)で説

明したように、オブジェクトを内包し且つ透視変換後のオブジェクトの大きさに応じてその大きさが変化するような形状の仮想ポリゴン(エフェクト領域)を生成し、その仮想ポリゴンに対してワークバッファ上のマッピング画像をマッピングする。

【0150】また、図19で描画された輪郭線EDLの画像が上書きにより消えてしまわないように、ソースαテストとデスティネーションαテストとを併用する。

【0151】即ち図22に示すように、

①テクスチャ座標はシフトさせず、

ソースαテスト(書き込み元であるマッピング画像に対するαテスト)でα>0のピクセルだけを合格にし、

デスティネーションαテスト(書き込み先であるフレームバッファの画像に対するαテスト)でα=αJのピクセルだけを合格にする。

【0152】上記の条件により、ワークバッファ上のマッピング画像が補間されずにそのままフレームバッファに描画される。

【0153】の条件により、マッピング画像(ワークバッファの画像)の中で、輪郭線色に設定されたピクセルについては描画が禁止される。

【0154】の条件により、図19で輪郭線の画像が描画されている0<α<αJのピクセルの領域に対する上書きが禁止される。即ち、オブジェクトOBの輪郭EDの内側領域から、OBの輪郭線EDL(具体的には右辺及び下辺のEDL)の領域を除いた領域に対してだけ、ワークバッファ上のぼけていないオブジェクトOBの画像が描画される。これにより、オブジェクトOBの全体がぼけた画像になってしまう問題を解消できる。

【0155】以上のように本実施形態の1つの特徴は、オブジェクトOBの輪郭EDの内側領域と外側領域とでα値を異ならせると共に(図17)、色のみならずα値もバイリニアフィルタにより補間し、補間後のα値に基づいて種々の判別を行っている点にある。即ち、補間後のα値に基づいて、オブジェクトOBの輪郭線EDLの領域(0<α<αJとなる領域)を判別したり、輪郭EDの内側領域から輪郭線EDLの領域を除いた領域を判別したりする。このようにすることで、少ない処理負担で、オブジェクトOBの輪郭線EDLを生成できるようになる。

【0156】3. 本実施形態の処理

次に、本実施形態の処理の詳細例について、図23、図24、図25のフローチャートを用いて説明する。

【0157】図23は、エフェクト領域の画像に対してだけ画像エフェクト処理を施す処理に関するフローチャートである。

【0158】まず、オブジェクトに対するジオメトリ処理を行い、オブジェクトをスクリーンに透視変換(アフィン変換)する(ステップS1)。

【0159】次に、透視変換後のオブジェクトの頂点の座標に基づき、オブジェクトの頂点のX、Y座標の最小値XMIN、YMIN、最大値XMAX、YMAXを求める(ステップS2)。

【0160】次に、求められた(XMIN、YMIN)、(XMAX、YMAX)に基づいて、図3

(A)、(B)で説明したように、エフェクト領域の形状の仮想ポリゴンを生成する(ステップS3)。この場合の仮想ポリゴンの頂点VVX1~VVX4は以下のようになる。

【0161】

VVX1 (XMIN、YMIN)

VVX2 (XMIN、YMAX)

VVX3 (XMAX、YMAX)

VVX4 (XMAX、YMIN)

次に、図7、図13、図16(A)~(C)で説明したようなエフェクト領域の画像(マッピング画像)を用いた仮想ポリゴンへのテクスチャマッピングを行うことで、エフェクト領域の画像に画像エフェクト処理を施す(ステップS4)。

【0162】なお、図19のオブジェクトOBの上辺、左辺にある輪郭線EDLのように、輪郭EDの外側領域に輪郭線EDLがはみ出す場合には、はみ出した輪郭線が適正に描画されるように、仮想ポリゴン(画像エフェクト領域)の大きさを上下左右方向に若干だけ大きくする。より具体的には、X、Y座標の最小値をXMIN、YMINとし、最大値をXMAX、YMAXとした場合に、XMIN、YMINから1ピクセルを減算してXMIN' = XMIN - 1、YMIN' = YMIN - 1を求めると共に、XMAX、YMAXに1ピクセルを加算してXMAX' = XMAX + 1、YMAX' = YMAX + 1を求める。そして、これらの求められた(XMIN'、YMIN')、(XMAX'、YMAX')に基づいて、仮想ポリゴンの頂点VVX1~VVX4を決めるようにする。

【0163】図24は、図5(A)、(B)で説明したように、オブジェクトの簡易オブジェクトを利用してエフェクト領域を設定し、そのエフェクト領域の画像に画像エフェクト処理を施す処理に関するフローチャートである。

【0164】図24が図23と異なるのは、ステップS11で、簡易オブジェクトに対する透視変換を行う点と、ステップS12で、透視変換後の簡易オブジェクトの頂点の座標に基づいて、(XMIN、YMIN)、(XMAX、YMAX)を求めている点であり、それ以外は同じ処理となる。

【0165】図25は、オブジェクトに輪郭線を付加する処理に関するフローチャートである。

【0166】まず、図17で説明したようにワークバッファを輪郭線の画像(RL、GL、BL、 α L)で初期

化する(ステップS21)。

【0167】次に、頂点の α 値が $\alpha = \alpha_J (> 0)$ に設定されたオブジェクトを透視変換して、ワークバッファに描画する(ステップS22)。その際に、ワークバッファ用のZバッファを用いて陰面消去を行うようにする。

【0168】次に、図18で説明したように、ワークバッファ上のマッピング画像をテクスチャ座標をシフトしてバイリニアフィルタ方式で仮想ポリゴンにマッピングしながら、その仮想ポリゴンをフレームバッファに描画する(ステップS23)。その際に、ソース α テストとして $\alpha > 0$ を指定する。また、オブジェクトの情報(例えば頂点座標、代表点の座標等)に基づき得られる擬似的なZ値を仮想ポリゴンに設定しておき、通常のZバッファを用いて陰面消去を行う。

【0169】次に、ワークバッファ上のマッピング画像を仮想ポリゴンにマッピングしながら、その仮想ポリゴンをフレームバッファに描画する(ステップS24)。その際に、ソース α テストとして $\alpha > 0$ を指定し、デスティネーション α テストとして $\alpha = \alpha_J$ を指定する。また、オブジェクトの情報に基づき得られる擬似的なZ値を仮想ポリゴンに設定しておき、通常のZバッファを用いて陰面消去を行う。

【0170】以上のようにして、オブジェクトに輪郭線を付加した画像(輪郭線を強調した画像)を得ることができる。

【0171】4. ハードウェア構成

次に、本実施形態を実現できるハードウェアの構成の一例について図26を用いて説明する。

【0172】メインプロセッサ900は、CD982(情報記憶媒体)に格納されたプログラム、通信インターフェース990を介して転送されたプログラム、或いはROM950(情報記憶媒体の1つ)に格納されたプログラムなどに基づき動作し、ゲーム処理、画像処理、音処理などの種々の処理を実行する。

【0173】コプロセッサ902は、メインプロセッサ900の処理を補助するものであり、高速並列演算が可能な積和算器や除算器を有し、マトリクス演算(ベクトル演算)を高速に実行する。例えば、オブジェクトを移動させたり動作(モーション)させるための物理シミュレーションに、マトリクス演算などの処理が必要な場合には、メインプロセッサ900上で動作するプログラムが、その処理をコプロセッサ902に指示(依頼)する。

【0174】ジオメトリプロセッサ904は、座標変換、透視変換、光源計算、曲面生成などのジオメトリ処理を行うものであり、高速並列演算が可能な積和算器や除算器を有し、マトリクス演算(ベクトル演算)を高速に実行する。例えば、座標変換、透視変換、光源計算などの処理を行う場合には、メインプロセッサ900で動

作するプログラムが、その処理をジオメトリプロセッサ904に指示する。

【0175】データ伸張プロセッサ906は、圧縮された画像データや音データを伸張するデコード処理を行ったり、メインプロセッサ900のデコード処理をアクセレートする処理を行う。これにより、オープニング画面、インターミッション画面、エンディング画面、或いはゲーム画面などにおいて、MPEG方式等で圧縮された動画像を表示できるようになる。なお、デコード処理の対象となる画像データや音データは、ROM950、CD982に格納されたり、或いは通信インターフェース990を介して外部から転送される。

【0176】描画プロセッサ910は、ポリゴンや曲面などのプリミティブ面で構成されるオブジェクトの描画（レンダリング）処理を高速に実行するものである。オブジェクトの描画の際には、メインプロセッサ900は、DMAコントローラ970の機能を利用して、オブジェクトデータを描画プロセッサ910に渡すと共に、必要であればテクスチャ記憶部924にテクスチャを転送する。すると、描画プロセッサ910は、これらのオブジェクトデータやテクスチャに基づいて、Zバッファなどを利用した陰面消去を行いながら、オブジェクトをフレームバッファ922に高速に描画する。また、描画プロセッサ910は、 α ブレンディング（半透明処理）、デプスクーイング、ミップマッピング、フォグ処理、バイリニア・フィルタリング、トライリニア・フィルタリング、アンチエイリアシング、シェーディング処理なども行うことができる。そして、1フレーム分の画像がフレームバッファ922に書き込まれると、その画像はディスプレイ912に表示される。

【0177】サウンドプロセッサ930は、多チャンネルのADPCM音源などを内蔵し、BGM、効果音、音声などの高品質のゲーム音を生成する。生成されたゲーム音は、スピーカ932から出力される。

【0178】ゲームコントローラ942からの操作データや、メモ리카ード944からのセーブデータ、個人データは、シリアルインターフェース940を介してデータ転送される。

【0179】ROM950にはシステムプログラムなどが格納される。なお、業務用ゲームシステムの場合には、ROM950が情報記憶媒体として機能し、ROM950に各種プログラムが格納されることになる。なお、ROM950の代わりにハードディスクを利用するようにしてもよい。

【0180】RAM960は、各種プロセッサの作業領域として用いられる。

【0181】DMAコントローラ970は、プロセッサ、メモリ（RAM、VRAM、ROM等）間でのDMA転送を制御するものである。

【0182】CDドライブ980は、プログラム、画像

データ、或いは音データなどが格納されるCD982

（情報記憶媒体）を駆動し、これらのプログラム、データへのアクセスを可能にする。

【0183】通信インターフェース990は、ネットワークを介して外部との間でデータ転送を行うためのインターフェースである。この場合に、通信インターフェース990に接続されるネットワークとしては、通信回線（アナログ電話回線、ISDN）、高速シリアルバスなどを考えることができる。そして、通信回線を利用することでインターネットを介したデータ転送が可能になる。また、高速シリアルバスを利用することで、他のゲームシステムとの間でのデータ転送が可能になる。

【0184】なお、本発明の各手段は、その全てを、ハードウェアのみにより実行してもよいし、情報記憶媒体に格納されるプログラムや通信インターフェースを介して配信されるプログラムのみにより実行してもよい。或いは、ハードウェアとプログラムの両方により実行してもよい。

【0185】そして、本発明の各手段をハードウェアとプログラムの両方により実行する場合には、情報記憶媒体には、本発明の各手段をハードウェアを利用して実行するためのプログラムが格納されることになる。より具体的には、上記プログラムが、ハードウェアである各プロセッサ902、904、906、910、930等に処理を指示すると共に、必要であればデータを渡す。そして、各プロセッサ902、904、906、910、930等は、その指示と渡されたデータとに基づいて、本発明の各手段を実行することになる。

【0186】図27（A）に、本実施形態を業務用ゲームシステムに適用した場合の例を示す。プレーヤは、ディスプレイ1100上に映し出されたゲーム画像を見ながら、レバー1102、ボタン1104等を操作してゲームを楽しむ。内蔵されるシステムボード（サーキットボード）1106には、各種プロセッサ、各種メモリなどが実装される。そして、本発明の各手段を実行するための情報（プログラム又はデータ）は、システムボード1106上の情報記憶媒体であるメモリ1108に格納される。以下、この情報を格納情報と呼ぶ。

【0187】図27（B）に、本実施形態を家庭用のゲームシステムに適用した場合の例を示す。プレーヤはディスプレイ1200に映し出されたゲーム画像を見ながら、ゲームコントローラ1202、1204を操作してゲームを楽しむ。この場合、上記格納情報は、本体システムに着脱自在な情報記憶媒体であるCD1206、或いはメモ리카ード1208、1209等に格納されている。

【0188】図27（C）に、ホスト装置1300と、このホスト装置1300とネットワーク1302（LANのような小規模ネットワークや、インターネットのような広域ネットワーク）を介して接続される端末130

4-1~1304-nとを含むシステムに本実施形態を適用した場合の例を示す。この場合、上記格納情報は、例えばホスト装置1300が制御可能な磁気ディスク装置、磁気テープ装置、メモリ等の情報記憶媒体1306に格納されている。端末1304-1~1304-nが、スタンドアロンでゲーム画像、ゲーム音を生成できるものである場合には、ホスト装置1300からは、ゲーム画像、ゲーム音を生成するためのゲームプログラム等が端末1304-1~1304-nに配送される。一方、スタンドアロンで生成できない場合には、ホスト装置1300がゲーム画像、ゲーム音を生成し、これを端末1304-1~1304-nに伝送し端末において出力することになる。

【0189】なお、図27(C)の構成の場合に、本発明の各手段を、ホスト装置(サーバー)と端末とで分散して実行するようにしてもよい。また、本発明の各手段を実行するための上記格納情報を、ホスト装置(サーバー)の情報記憶媒体と端末の情報記憶媒体に分散して格納するようにしてもよい。

【0190】またネットワークに接続する端末は、家庭用ゲームシステムであってもよいし業務用ゲームシステムであってもよい。そして、業務用ゲームシステムをネットワークに接続する場合には、業務用ゲームシステムとの間で情報のやり取りが可能であると共に家庭用ゲームシステムとの間でも情報のやり取りが可能なセーブ用情報記憶装置(メモリカード、携帯型ゲーム装置)を用いることが望ましい。

【0191】なお本発明は、上記実施形態で説明したものに限らず、種々の変形実施が可能である。

【0192】例えば、本発明のうち従属請求項に係る発明においては、従属先の請求項の構成要件の一部を省略する構成とすることもできる。また、本発明の1の独立請求項に係る発明の要部を、他の独立請求項に従属させることもできる。

【0193】また、エフェクト領域の設定手法は図3(A)、(B)、図4(A)、(B)、図5(A)、(B)で説明した手法が特に望ましいが、これに限定されず種々の変形実施が可能である。

【0194】また、エフェクト領域の画像に対して施す画像エフェクト処理も、本実施形態で説明したものが特に望ましいが、これに限定されるものではない。

【0195】また、インデックスカラー・テクスチャマッピング用のルックアップテーブルのインデックス番号として設定される画像情報としては、本実施形態で説明した情報が特に望ましいが、これに限定されるものではない。

【0196】また、テクセル補間方式のテクスチャマッピングとしてはバイリニアフィルタ方式のテクスチャマッピングが特に望ましいが、これに限定されるものではない。

【0197】また、本発明は種々のゲーム(格闘ゲー

ム、シューティングゲーム、ロボット対戦ゲーム、スポーツゲーム、競争ゲーム、ロールプレイングゲーム、音楽演奏ゲーム、ダンスゲーム等)に適用できる。

【0198】また本発明は、業務用ゲームシステム、家庭用ゲームシステム、多数のプレーヤが参加する大型アトラクションシステム、シミュレータ、マルチメディア端末、ゲーム画像を生成するシステムボード等の種々のゲームシステム(画像生成システム)に適用できる。

【図面の簡単な説明】

【図1】本実施形態のゲームシステムの機能ブロック図の例である。

【図2】本実施形態により生成されるゲーム画像の例である。

【図3】図3(A)、(B)は、エフェクト領域の設定手法について説明するための図である。

【図4】図4(A)、(B)は、透視変換後のオブジェクトの頂点の座標に基づいてエフェクト領域を設定する手法について説明するための図である。

【図5】図5(A)、(B)は、透視変換後の簡易オブジェクトの頂点の座標に基づいてエフェクト領域を設定する手法について説明するための図である。

【図6】インデックスカラー・テクスチャマッピングについて説明するための図である。

【図7】インデックスカラー・テクスチャマッピング用のLUTを有効利用して、画像エフェクト処理を施す手法について示す図である。

【図8】図8(A)、(B)は、ガンマ補正の変換特性とガンマ補正用LUTの例について示す図である。

【図9】図9(A)、(B)は、ネガポジ反転の変換特性とネガポジ反転用LUTの例について示す図である。

【図10】図10(A)、(B)、(C)は、ポストリゼーション、ソラリゼーション、2値化の変換特性の例について示す図である。

【図11】ある色成分をインデックス番号に設定した場合に、他の色成分をマスクして、画像エフェクト処理後の画像を得る手法について説明するための図である。

【図12】バイリニアフィルタ方式のテクスチャマッピングについて説明するための図である。

【図13】バイリニアフィルタ方式のテクスチャマッピングを有効利用してぼかし画像を生成する手法について説明するための図である。

【図14】図14(A)、(B)は、バイリニアフィルタ方式の補間機能によりぼかし画像が生成される原理について説明するための図である。

【図15】図15(A)、(B)も、バイリニアフィルタ方式の補間機能によりぼかし画像が生成される原理について説明するための図である。

【図16】図16(A)、(B)、(C)は、ピクセル入れ替え処理について説明するための図である。

【図17】ワークパッド上に生成されたマッピング画

像の例について示すための図である。

【図18】マッピング画像をバイリニアフィルタ方式でテクスチャ座標をシフトさせながら仮想ポリゴンにマッピングしてフレームバッファに描画する手法について説明するための図である。

【図19】図18の手法によりフレームバッファ上に生成された画像の例について示す図である。

【図20】バイリニアフィルタ方式によりピクセルのRGB、 α 値を補間する手法について説明するための図である。

【図21】図21(A)、(B)は、オブジェクトの輪郭線を生成する手法について説明するための図である。

【図22】オブジェクトの全体がぼけた画像になるのを解消する手法について説明するための図である。

【図23】本実施形態の処理の詳細例について示すフローチャートである。

【図24】本実施形態の処理の詳細例について示すフローチャートである。

【図25】本実施形態の処理の詳細例について示すフローチャートである。

【図26】本実施形態を実現できるハードウェアの構成の一例を示す図である。

【図27】図27(A)、(B)、(C)は、本実施形態が適用される種々の形態のシステムの例を示す図である。

【符号の説明】

OB オブジェクト

SOB 簡易オブジェクト

EA エフェクト領域

ED 輪郭

EDL 輪郭線

VX1~VX6 頂点

VVX1~VVX4 頂点

VSX1~VSX8 頂点

100 処理部

110 ジオメトリ処理部

112 エフェクト領域設定部（仮想オブジェクト生成部）

114 インデックス番号設定部

120 描画部（オブジェクト・輪郭線描画部）

130 テクスチャマッピング部（画像エフェクト処理部）

132 合成部

134 陰面消去部

160 操作部

170 記憶部

172 主記憶部

174 描画領域

176 テクスチャ記憶部

178 LUT記憶部

180 情報記憶媒体

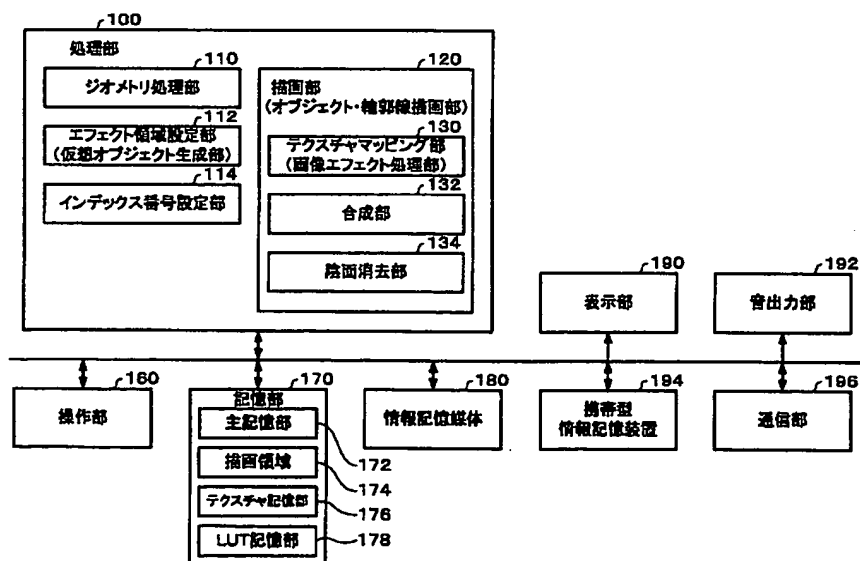
190 表示部

192 音出力部

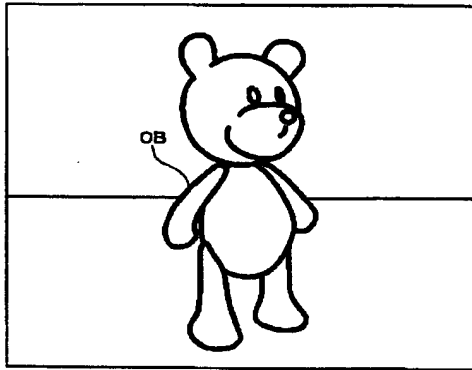
194 携帯型情報記憶装置

196 通信部

【図1】

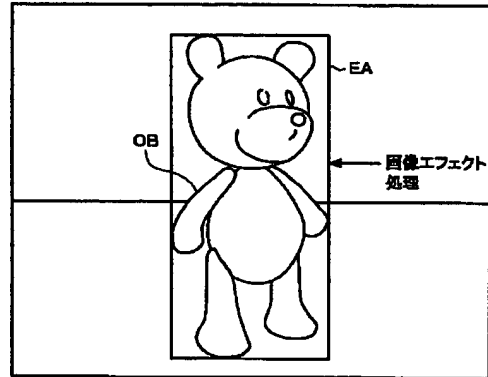


【図2】

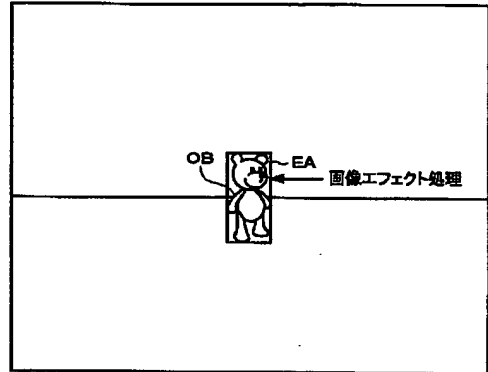


【図3】

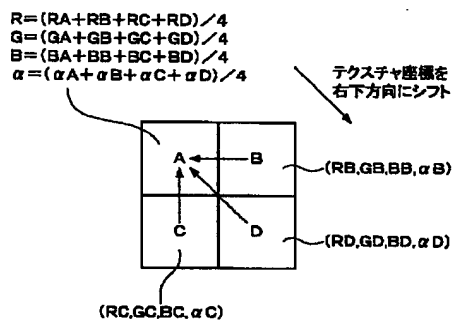
(A) 視点からの距離が近い場合



(B) 視点からの距離が遠い場合

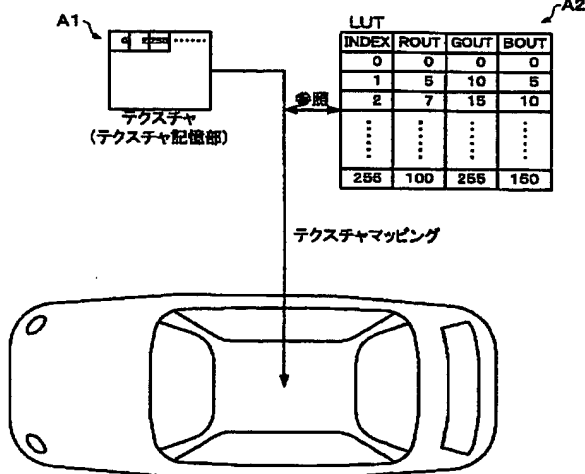


【図20】



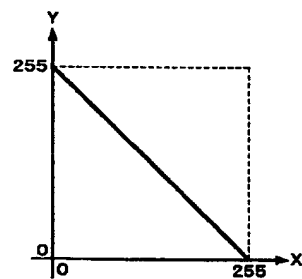
【図6】

インデックスカラー・テクスチャマッピング



【図9】

(A) ネガボジ反転

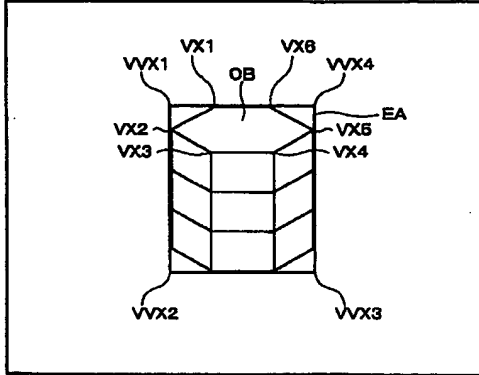


(B) ネガボジ反転用LUT

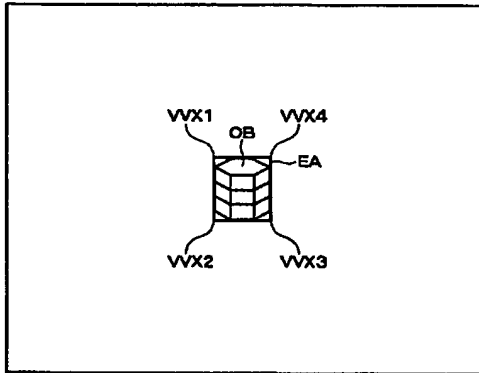
INDEX	ROUT	GOUT	BOUT
0	255	255	255
1	254	254	254
2	253	253	253
3	252	252	252
...
253	2	2	2
254	1	1	1
255	0	0	0

【図4】

(A) 視点からの距離が近い場合

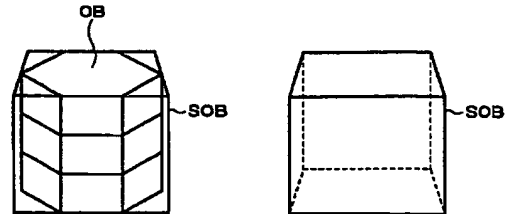


(B) 視点からの距離が遠い場合

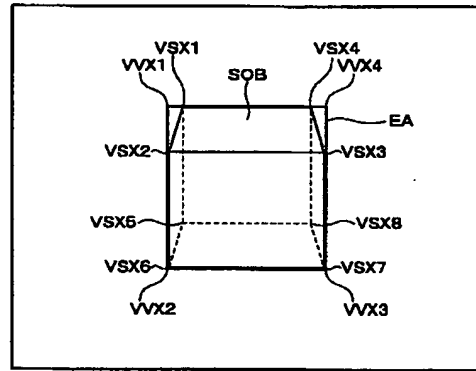


【図5】

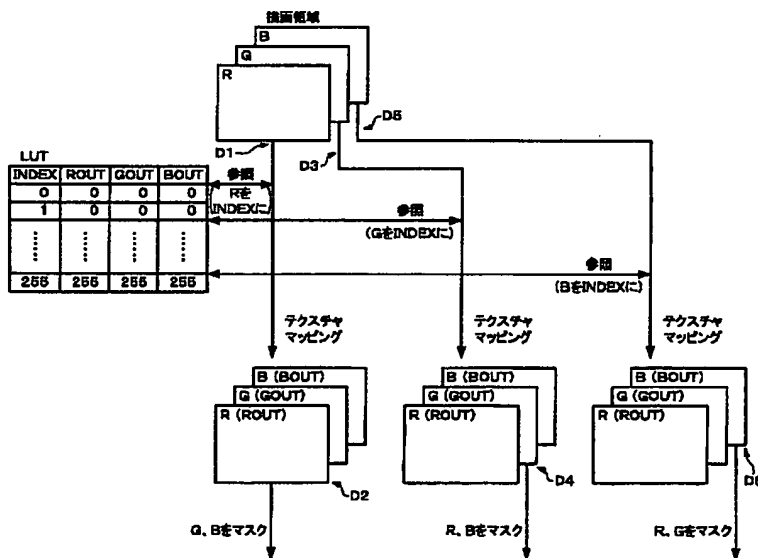
(A)



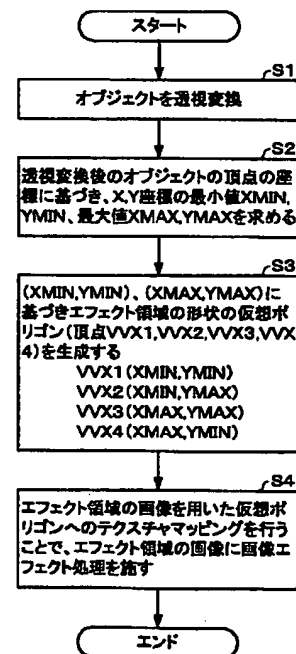
(B)



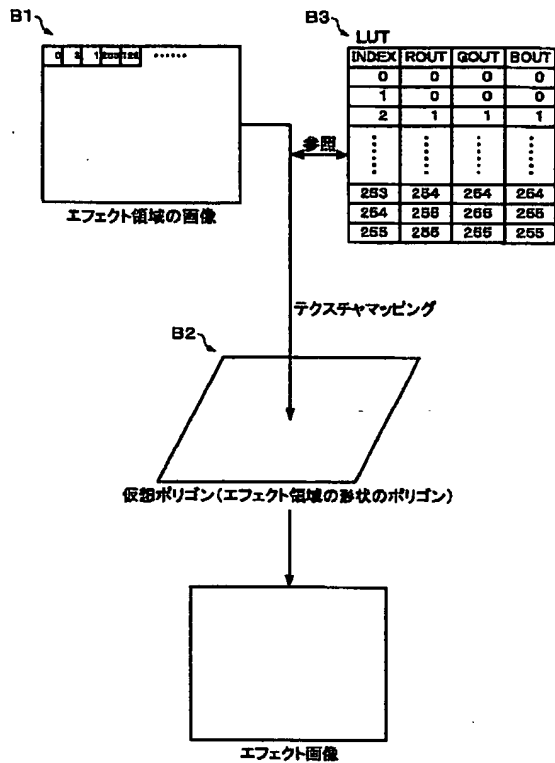
【図11】



【図23】



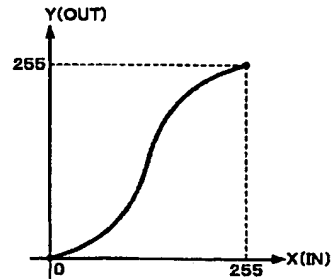
【図7】



【図8】

(A)

ガンマ補正

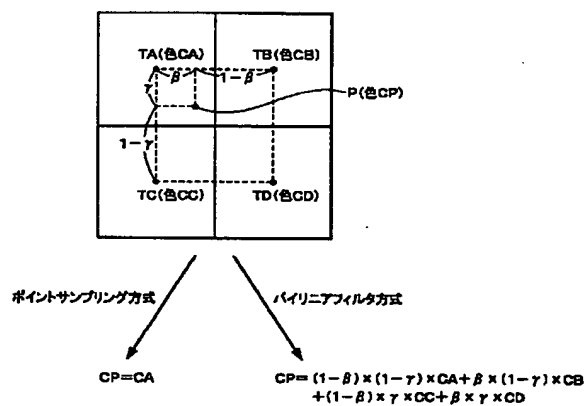


(B)

ガンマ補正用LUT

INDEX	ROUT	GOUT	BOUT
0	0	0	0
1	0	0	0
2	1	1	1
3	1	1	1
4	2	2	2
5	3	3	3
...
250	252	252	252
251	253	253	253
252	254	254	254
253	254	254	254
254	255	255	255
255	255	255	255

【図12】



【図16】

ピクセル入れ替え

(A)

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

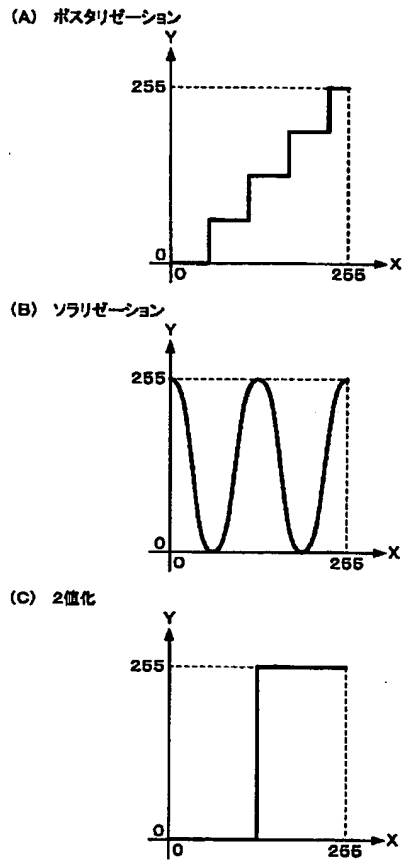
(B)

A	B	C	D	E
F	G	R	I	J
K	L	M	N	O
P	Q	H	S	T
U	V	W	X	Y

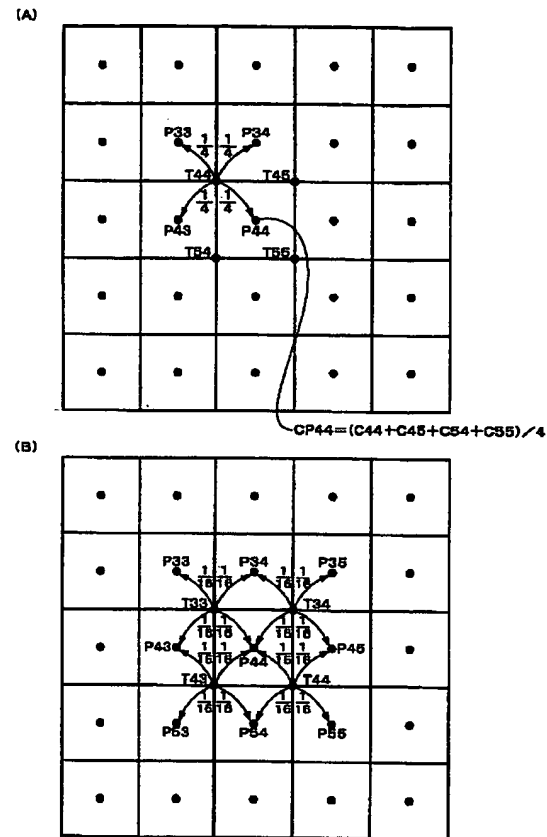
(C)

A	B	C	D	E
F	G	R	I	Q
K	L	M	N	O
P	J	H	S	T
U	V	W	X	Y

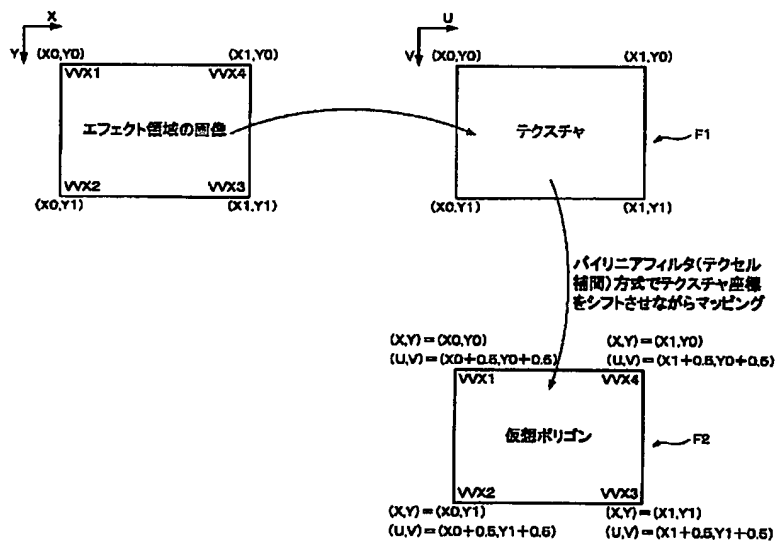
【図10】



【図14】



【図13】



【図15】

(A)

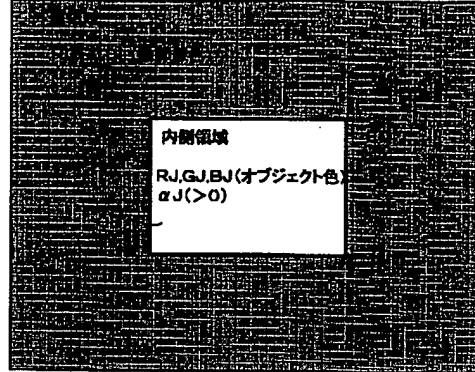
	$\frac{1}{16}$	$\frac{2}{16}$	$\frac{1}{16}$	
	$\frac{2}{16}$	$\frac{4}{16}$	$\frac{2}{16}$	
	$\frac{1}{16}$	$\frac{2}{16}$	$\frac{1}{16}$	

(B)

$\frac{1}{256}$	$\frac{4}{256}$	$\frac{6}{256}$	$\frac{4}{256}$	$\frac{1}{256}$
$\frac{4}{256}$	$\frac{16}{256}$	$\frac{24}{256}$	$\frac{16}{256}$	$\frac{4}{256}$
$\frac{6}{256}$	$\frac{24}{256}$	$\frac{36}{256}$	$\frac{24}{256}$	$\frac{6}{256}$
$\frac{4}{256}$	$\frac{16}{256}$	$\frac{24}{256}$	$\frac{16}{256}$	$\frac{4}{256}$
$\frac{1}{256}$	$\frac{4}{256}$	$\frac{6}{256}$	$\frac{4}{256}$	$\frac{1}{256}$

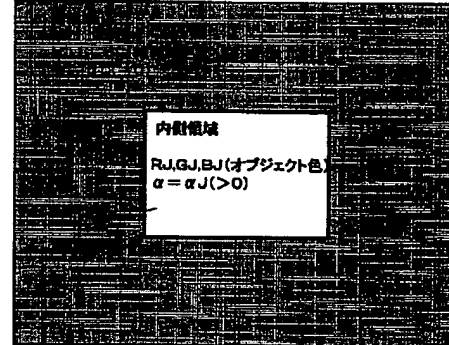
【図17】

ワークバッファ



【図18】

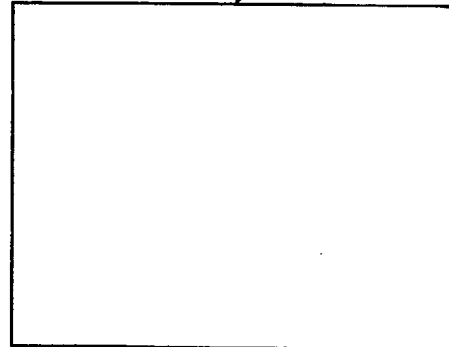
ワークバッファ



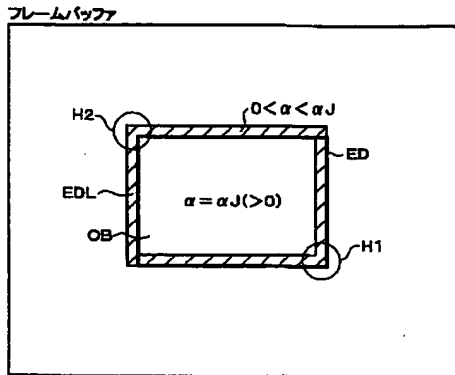
テクスチャ
マッピング

- ・バイリニアフィルタ(テクセル補間)方式
- ・テクスチャ座標を+0.5ピクセル(テクセル)だけシフト
- ・ソースαテストでα>0のピクセルだけを合格

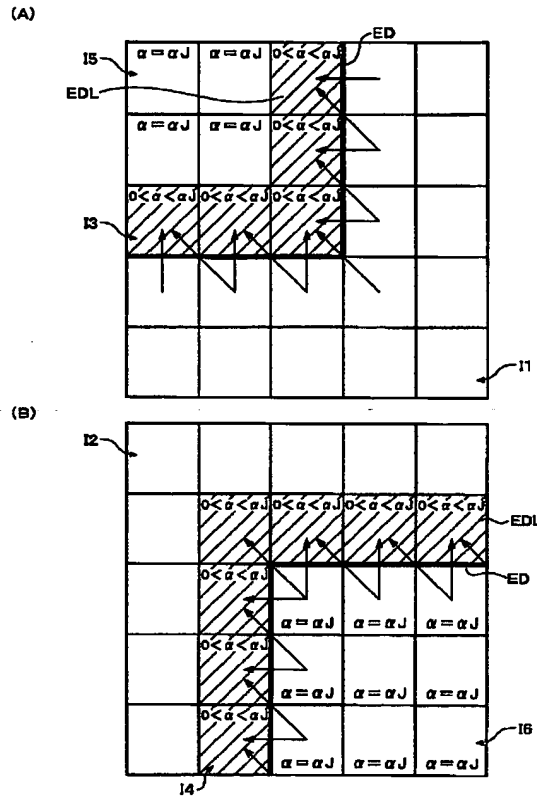
フレームバッファ



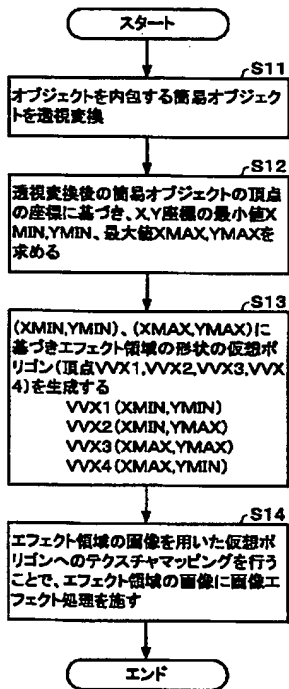
【図19】



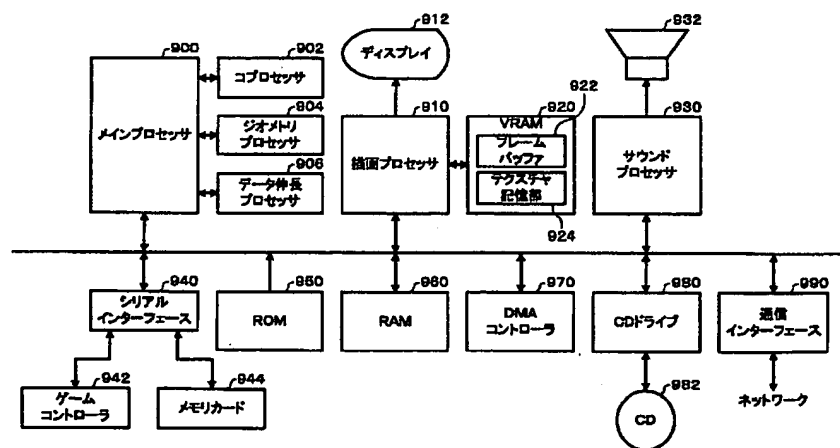
【図21】



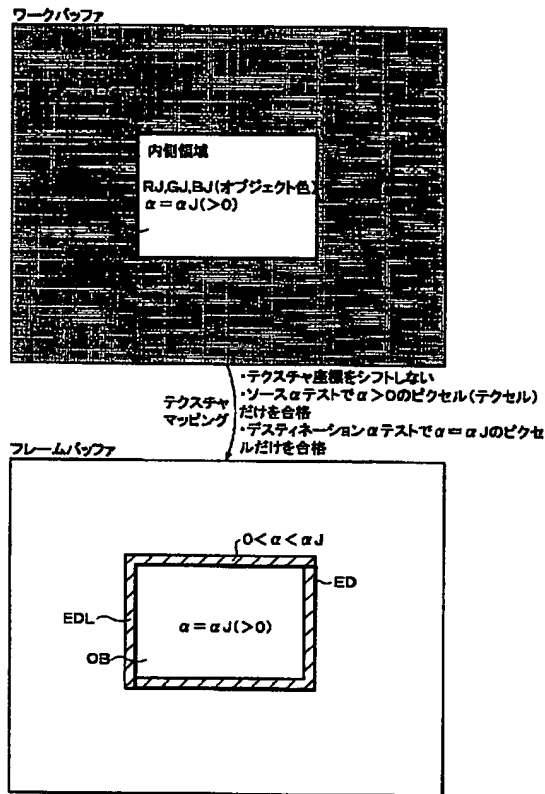
【図24】



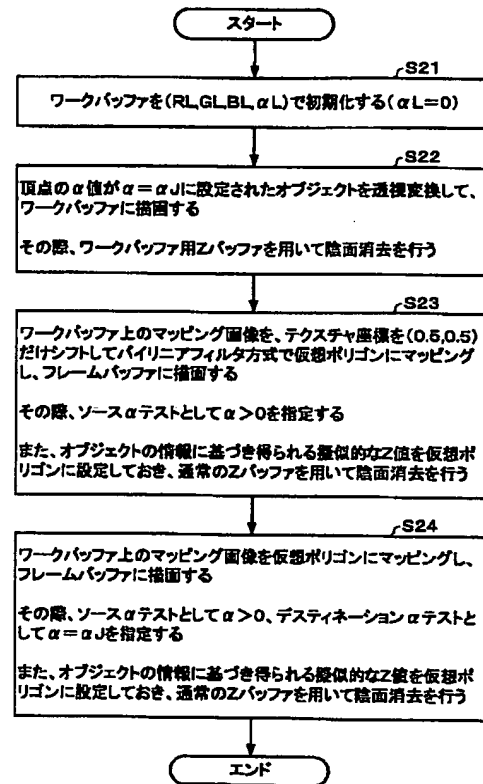
【図26】



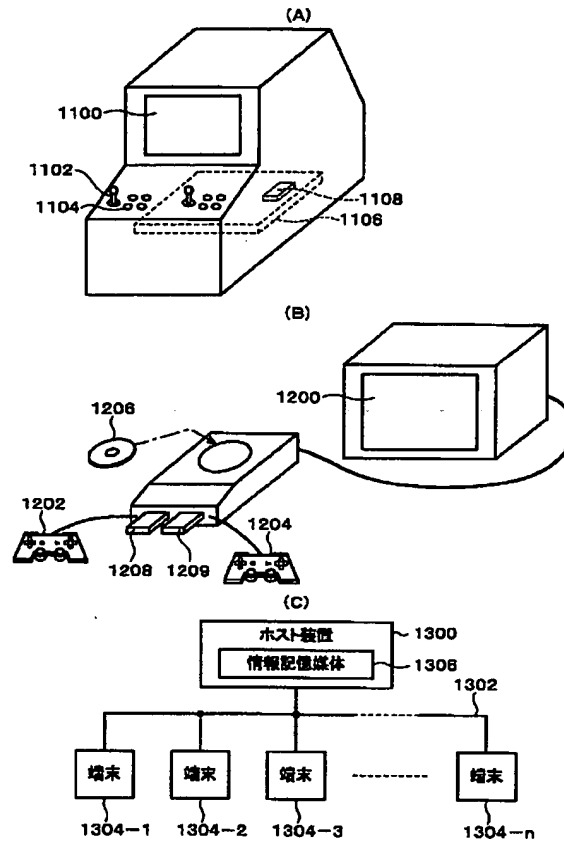
【図22】



【図25】



【図27】



フロントページの続き

Fターム(参考) 2C001 AA09 BA02 BC01 BC03 CA01
 CA06 CB01 CB04 CB06 CB08
 CC02 CC03 CC08
 5B050 BA08 BA09 EA06 EA11 EA28
 EA30 FA02
 5B080 AA13 BA04 FA02 FA03 GA11
 GA22